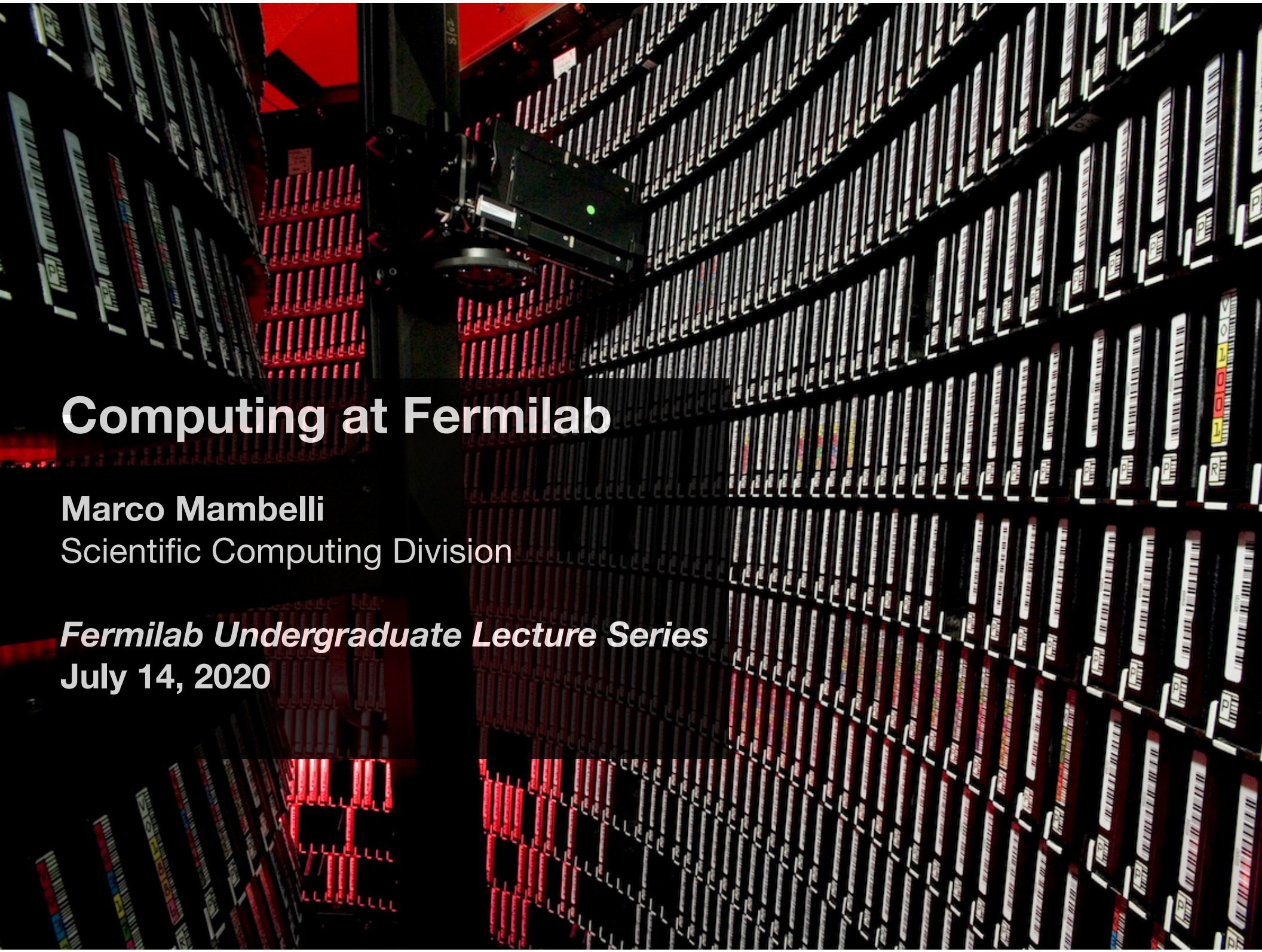


Computing at Fermilab

Marco Mambelli
Scientific Computing Division

Fermilab Undergraduate Lecture Series
July 14, 2020

Outline

- What is **scientific computing?**
  - How do we use computers to produce **science** from our **detectors**

- What is the **hardware** we need to accomplish this?
  - CPU
  - Storage
  - Network

- What is the **software** we need to accomplish this?
  - Reconstruction
  - Simulation
  - Analysis

- What is the **future** of scientific computing in HEP?

What is scientific computing?

Computational science

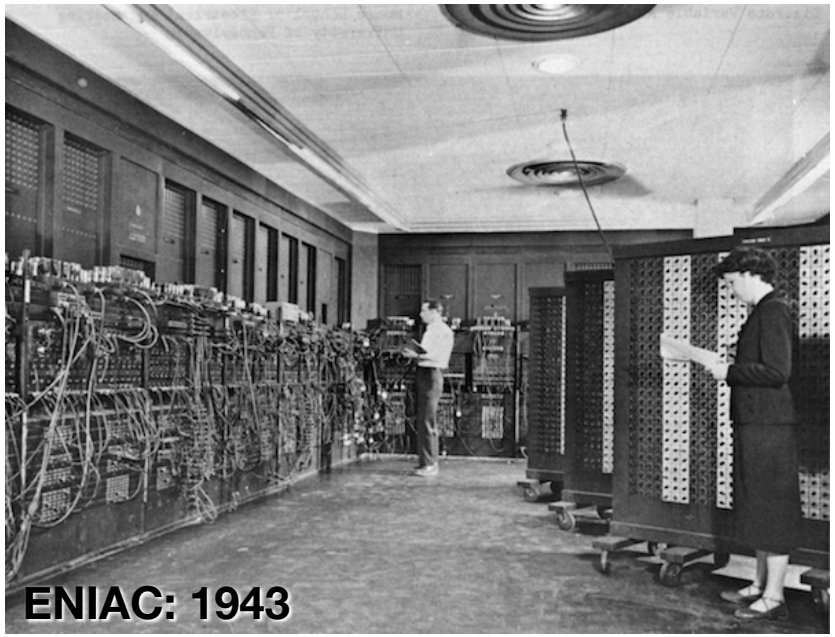
From Wikipedia, the free encyclopedia

Not to be confused with computer science.

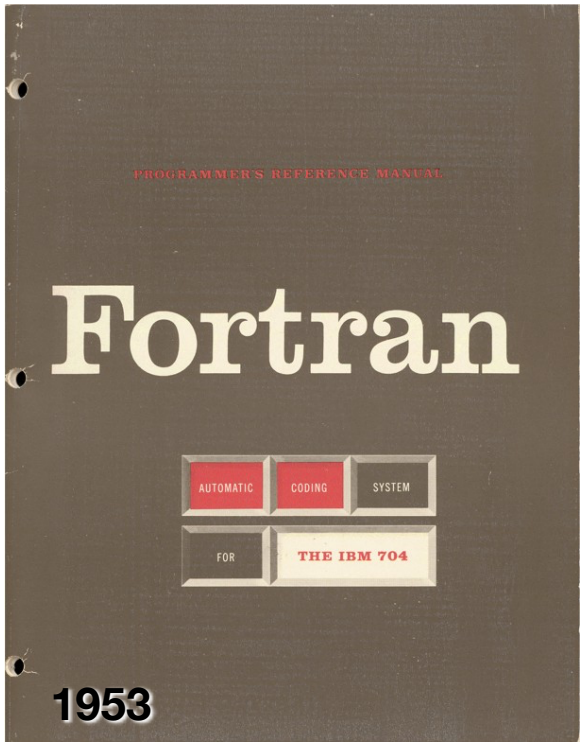
Computational science (also **scientific computing** or **scientific computation** (**SC**)) is a rapidly growing multidisciplinary field that uses advanced computing capabilities to understand and solve complex problems. It is an area of science which spans many disciplines, but at its core it involves the development of models and simulations to understand natural systems.

- Algorithms (numerical and non-numerical): mathematical models, computational models, and computer simulations developed to solve science (e.g., biological, physical, and social), engineering, and humanities problems
- Computer and information science that develops and optimizes the advanced system hardware, software, networking, and data management components needed to solve computationally demanding problems
- The computing infrastructure that supports both the science and engineering problem solving and the developmental computer and information science

In practical use, it is typically the application of computer simulation and other forms of computation from numerical analysis and theoretical computer science to solve problems in various scientific disciplines. The field is different from theory and laboratory experiment which are the traditional forms of science and engineering. The scientific

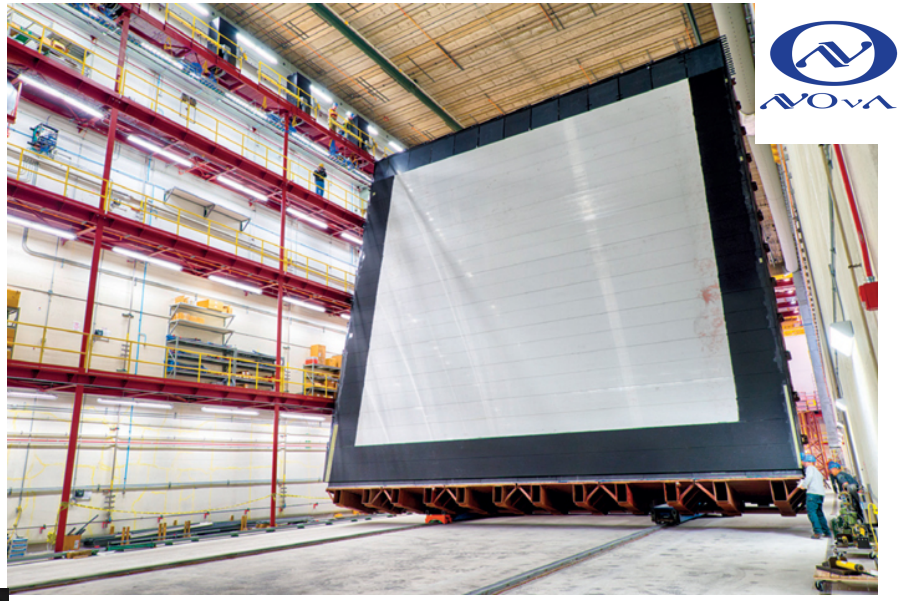


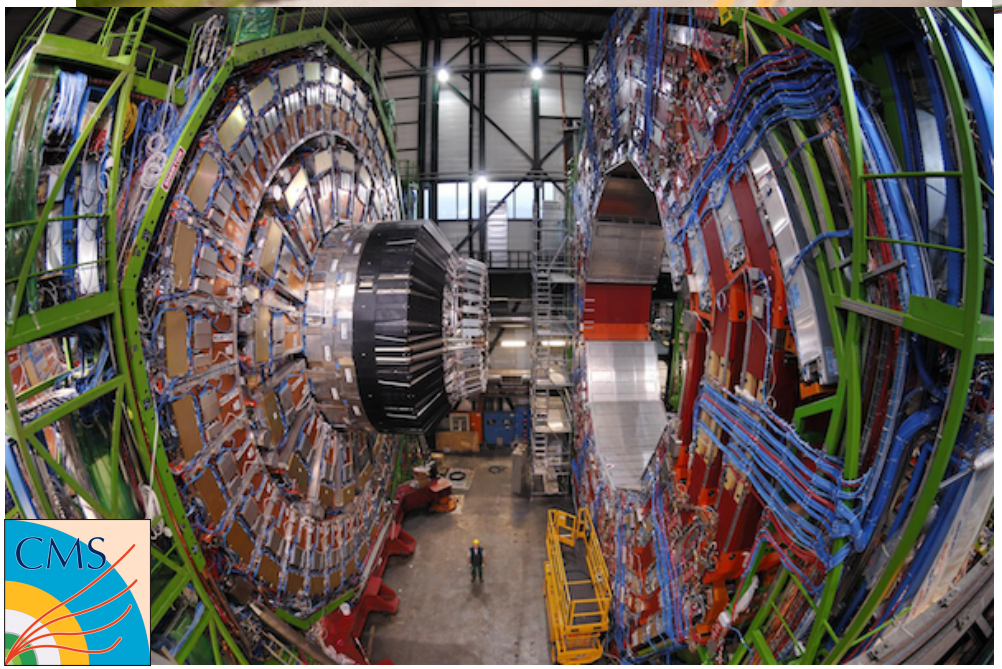
ENIAC: 1943

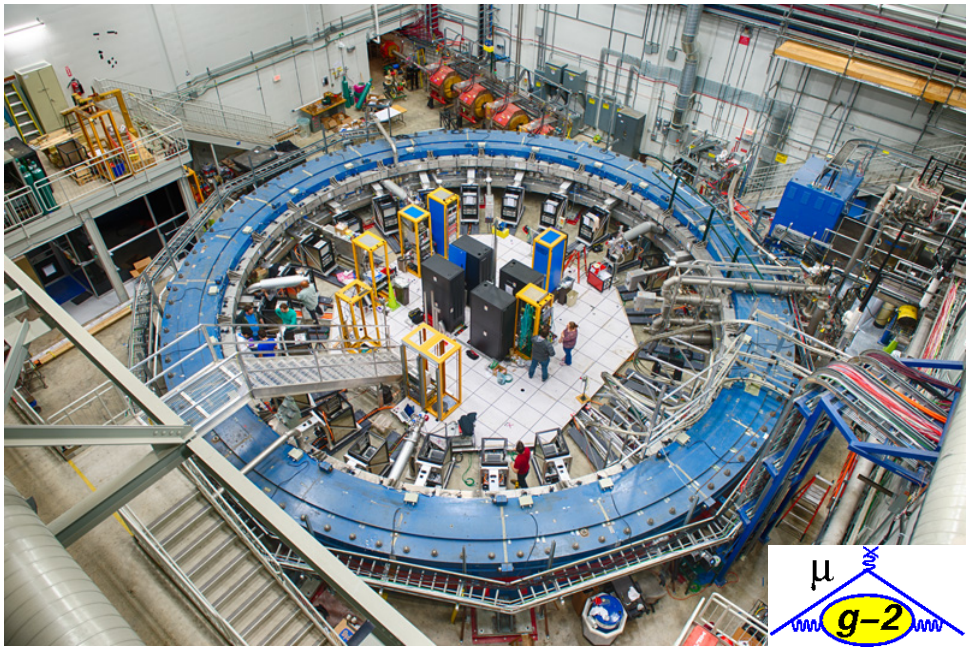


1953

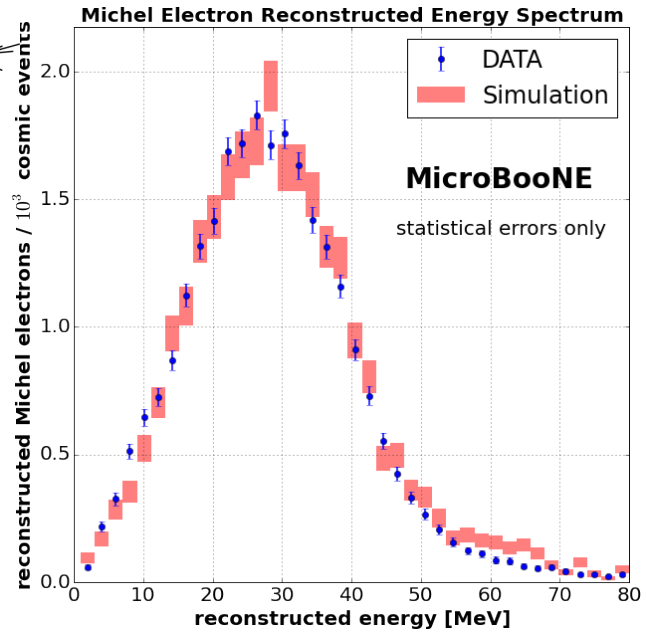
Scientific Computing in HEP: Getting from here…

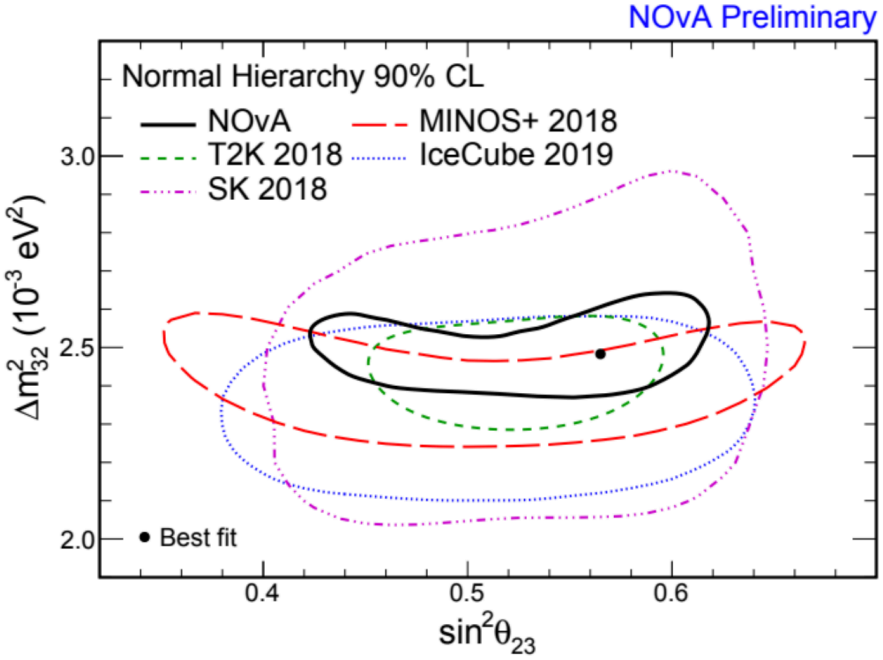


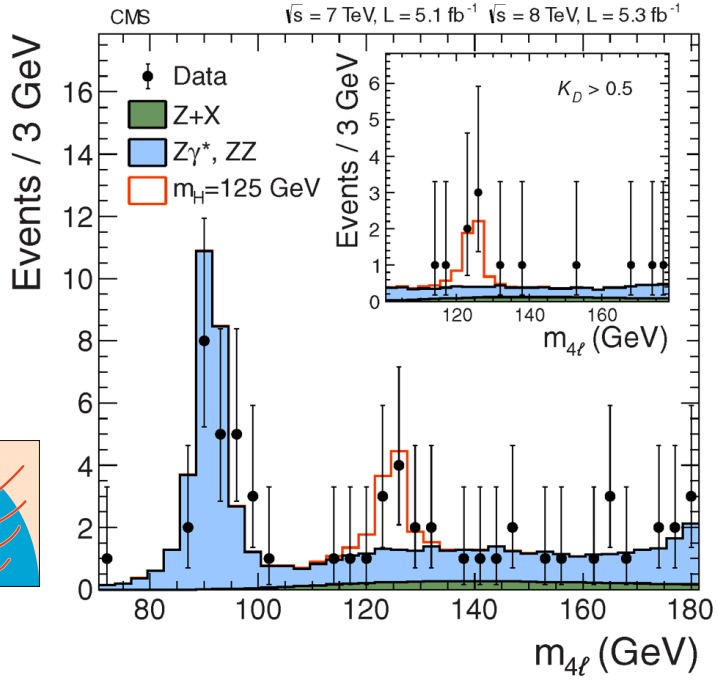


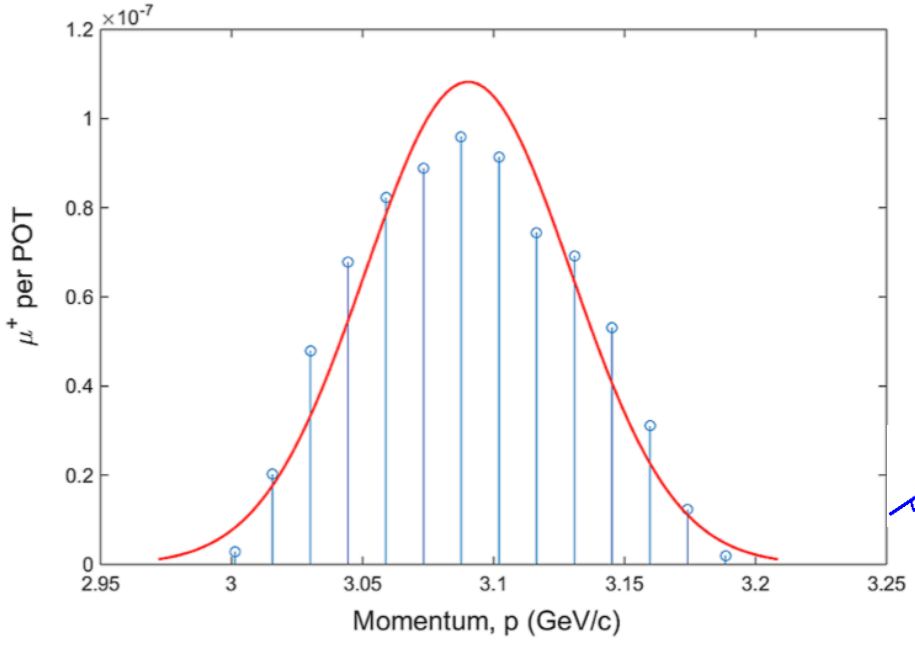


…to here

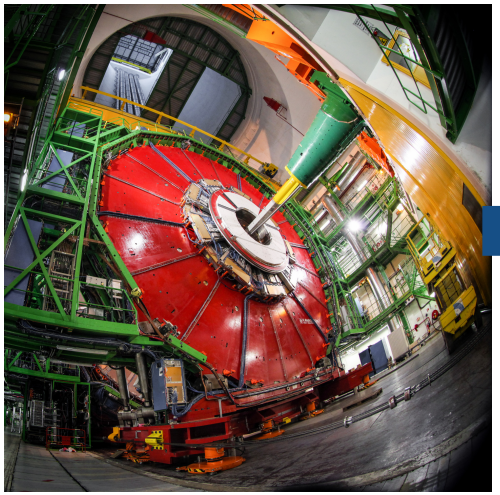

Michel Electron Reconstructed Energy Spectrum
DATA
Simulation
MicroBooNE
statistical errors only
reconstructed Michel electrons / 10^3 cosmic events
reconstructed energy [MeV]


NOvA Preliminary
Normal Hierarchy 90% CL
NOvA
MINOS+ 2018
T2K 2018
IceCube 2019
SK 2018
Best fit
Δm^2_{32} (10^{-3} eV^2)
$\sin^2\theta_{23}$


CMS
$\sqrt{s}$ = 7 TeV, L = 5.1 fb^{-1} $\sqrt{s}$ = 8 TeV, L = 5.3 fb^{-1}
Data
Z+X
Zγ*, ZZ
m_H=125 GeV
K_D > 0.5
Events / 3 GeV
$m_{4\ell}$ (GeV)


$\times 10^{-7}$
μ^+ per POT
Momentum, p (GeV/c)

The process



DAQ, Triggering

"Raw" data

Dist. CPU

Monte Carlo simulation

Dist. CPU

reco. software

Dist. CPU

Reconstructed data

database

Reduced data

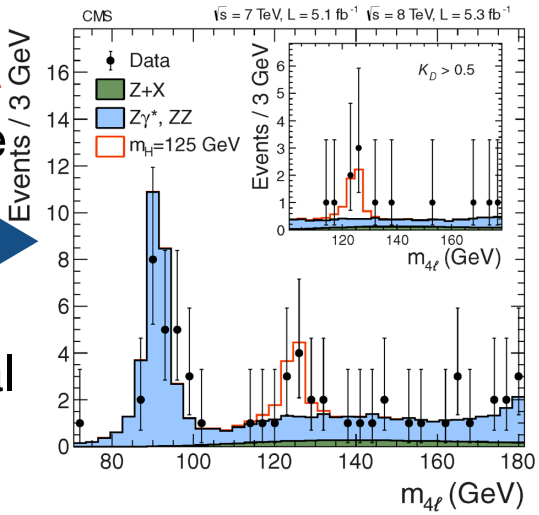
analysis software

(further) Reduced data

analysis software

Dist/local CPU

Dist/local CPU



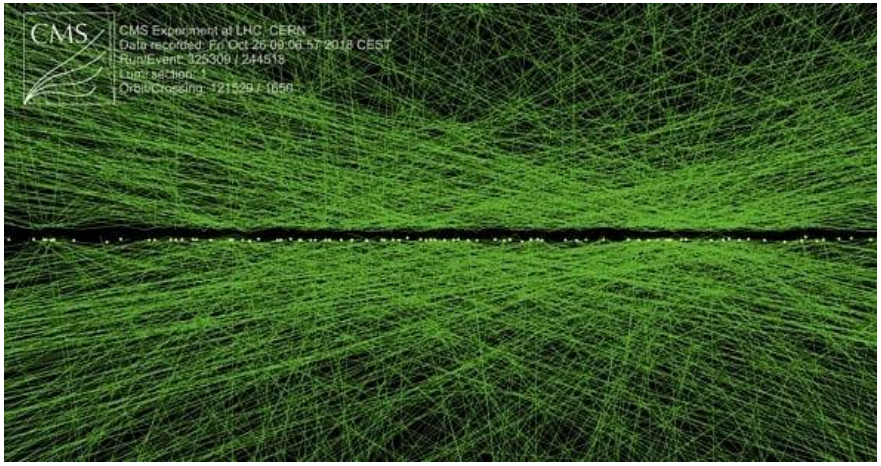
Scientific Computing at Fermilab

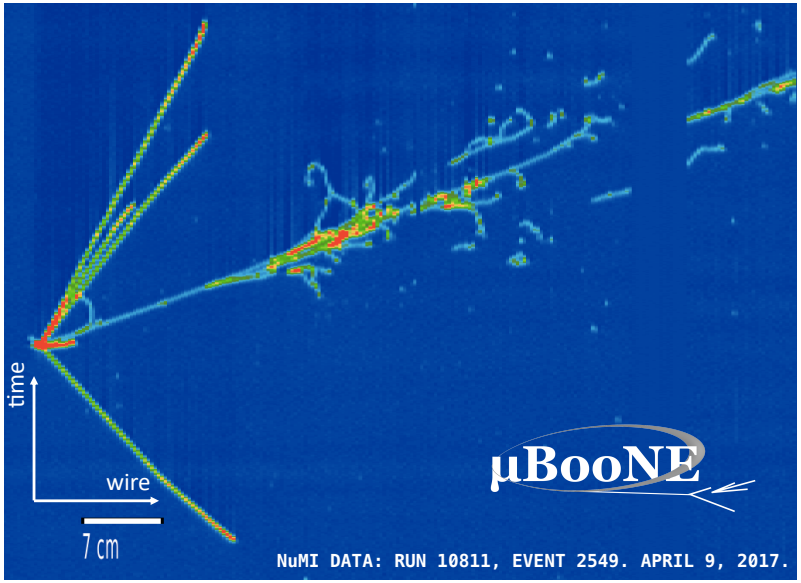
- The **computational effort** (hardware, software) required to turn **detector bits** into **scientific results**
  - At Fermilab, the **Scientific Computing Division** supports this
  - This is what I'll be focusing on
- A lot of critical computing is **not** part of this
  - Email, web, productivity tools, and associated infrastructure
  - At Fermilab, the **Core Computing Division** supports this
- **Quantum computing** will not be covered
  - See Henry Lamm's lecture on 7/2
- Some caveats
  - This is, at best, a **10,000' view**
  - I am a Computer Engineer working on the infrastructure that make scientific computing possible

Hardware

CPU

- **Reconstruction** and **simulation** of events are biggest CPU drivers
- Such computing is **“pleasantly parallel”**
  - Processing one event is completely independent of processing any other
  - Relatively short processing times but **many events** and **growing complexity**
- **CMS -** typical collider experiment
  - ~30 s/event (~30x more in a decade!)
  - ~billions events (simulated+collision)/ month
- **MicroBooNE -** liquid Argon (LAr) neutrino experiment
  - ~1-2 min/event
  - ~million events (simulated+beam)/month
  - 1 event in DUNE will have ~50x more channels (!)





Divide et Impera

- Julius Caesar

- Split the complex problem
- Solve the parts
- Get the overall solution

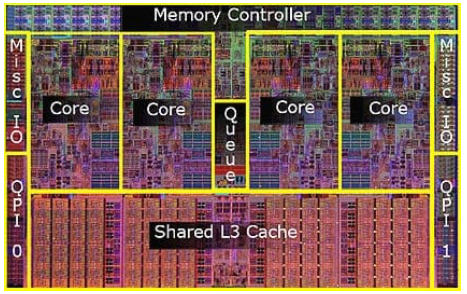
- Divide and Conquer

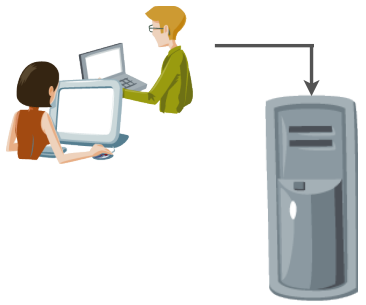
Divide and Conquer

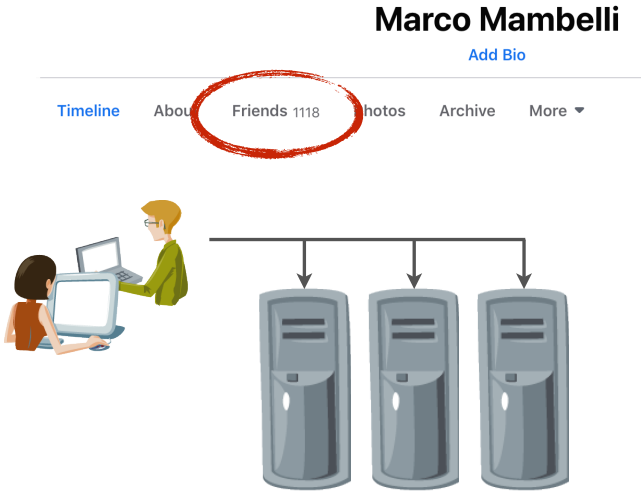
- CPU: Central Processing Unit typically refers to the whole chip
  - Most modern CPUs contain between 2 and 32 individual **cores**
  - Each core can process one instruction at a time
- Use one computer?

  30s/8 cores*1B events ~120 years

- Use your friends' computers as well
  - To get 1B events in one month, we require **1,440 8-core computers**
    - We are almost there with the friends!
  - Your **software and data** would need to get to each of those computers
  - You'd need to **collect output** from each
  - And you'd need **user accounts** on all of them
- Solution: find an easier way to get from one computer to many

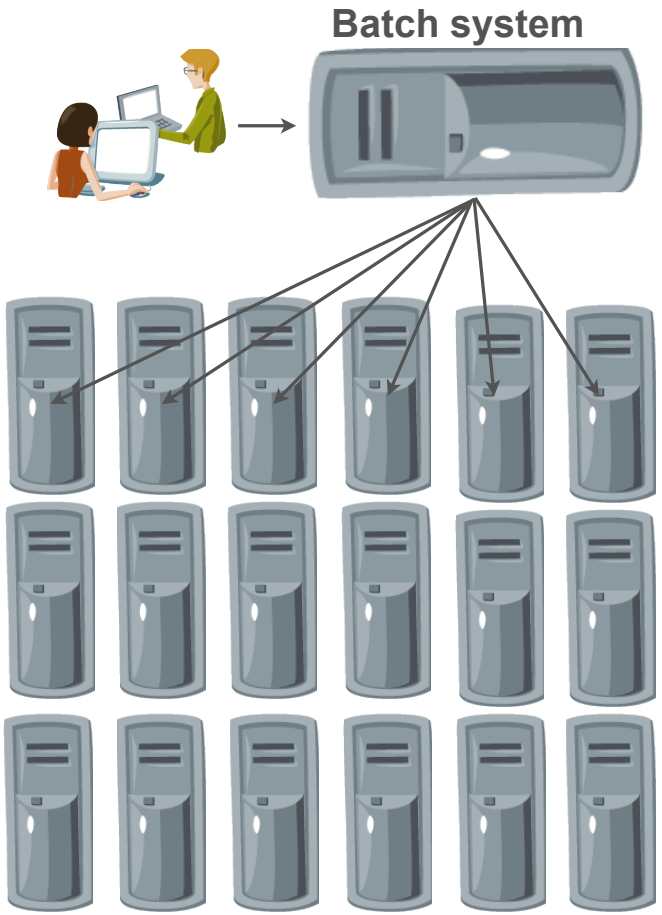






Divide and conquer: batch systems

- **Batch systems** allow just that
  - **Single entry point** manages a work queue (for many users)
  - Jobs are directed to **any available slot**
  - **Output** from each job handled in the same way
  - Batch system can handle user authentication on each individual computer
- A form of **high-throughput computing** (HTC)

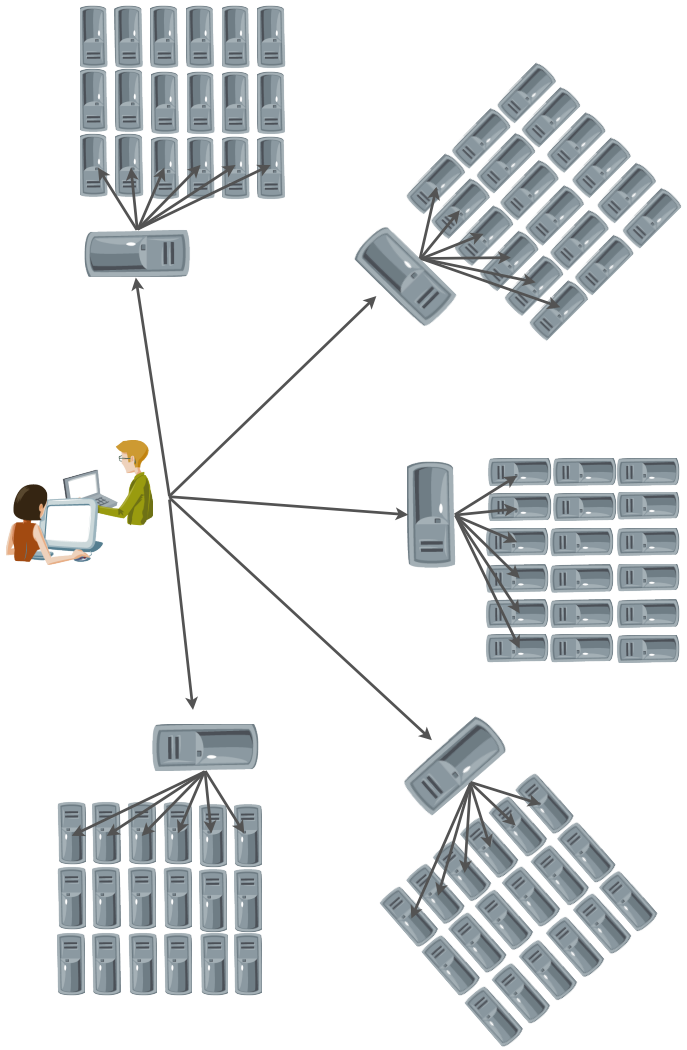


Some batch systems at Fermilab:
Fermigrid (~20k CPU cores)
US CMS Tier1 (~20k CPU cores)
CMS LPC (~5k CPU cores)

30s/20,000 cores*1B events ~17 days

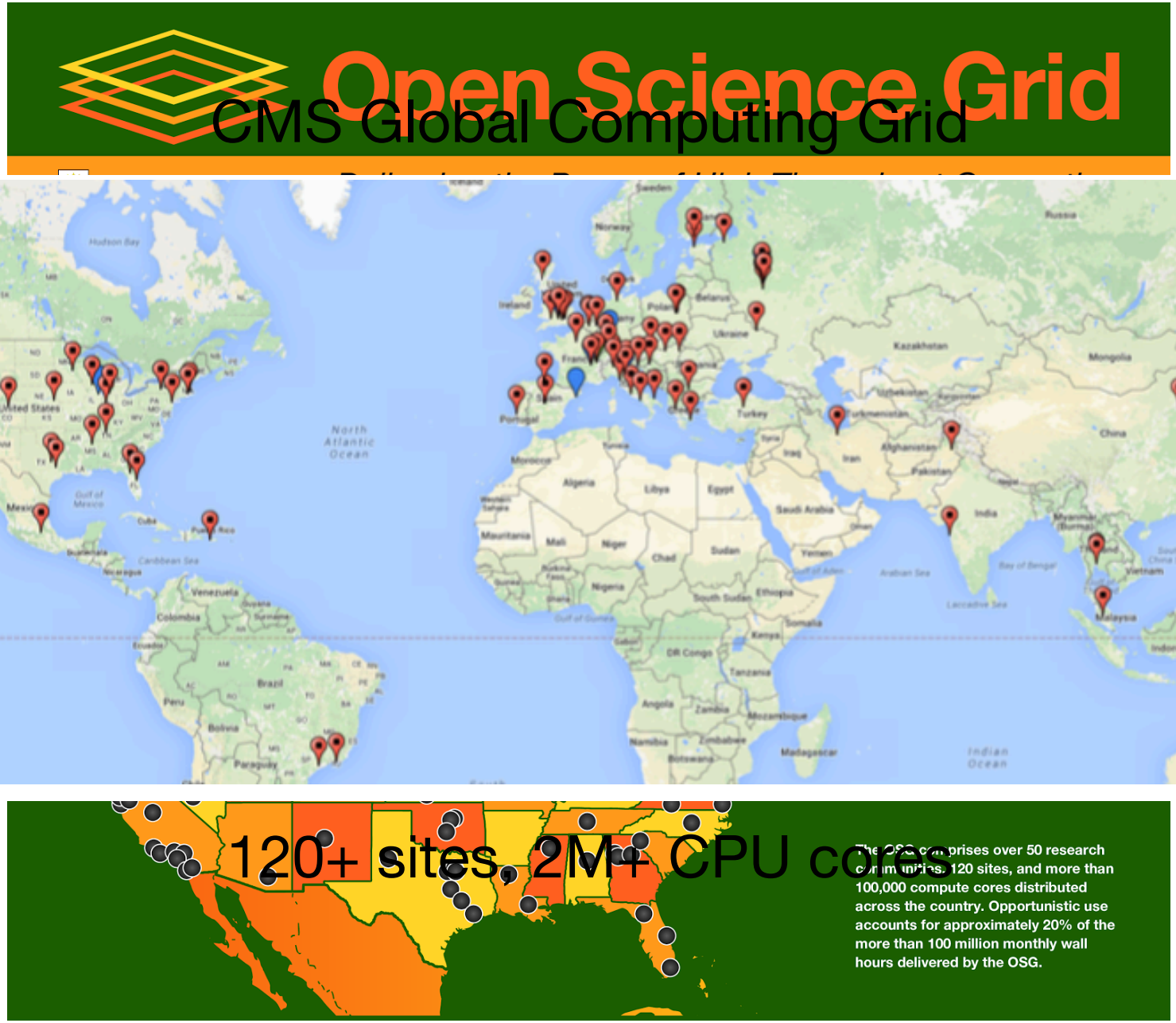
Divide and conquer: the Grid

- Even better yet, have many batch systems accessible from one point: a computing **grid**
  - Can incorporate **grid sites** (batch systems) from universities and labs across the world into one grid
  - **Distributed high-throughput computing** (DHTC)
- Analogy: utility grids



- Problem: **user account** commonality
- Solution: **tokens** or **grid certificates** and **Virtual Organization** (VO) trust model
  - A certificate is an encrypted “signature” that verifies you belong to an organization (e.g., a collaboration like NOvA)
  - Each site decides which VOs to trust

Overlapping grids



Divide and conquer: the Cloud

- And when you still don't have enough, then you can rent it
  - **Commercial Clouds** like AWS (Amazon), GCE (Google) and Azure (Microsoft) can rent you a seemly endless amount of computing power
  - **Elastic computing** because it expands at will

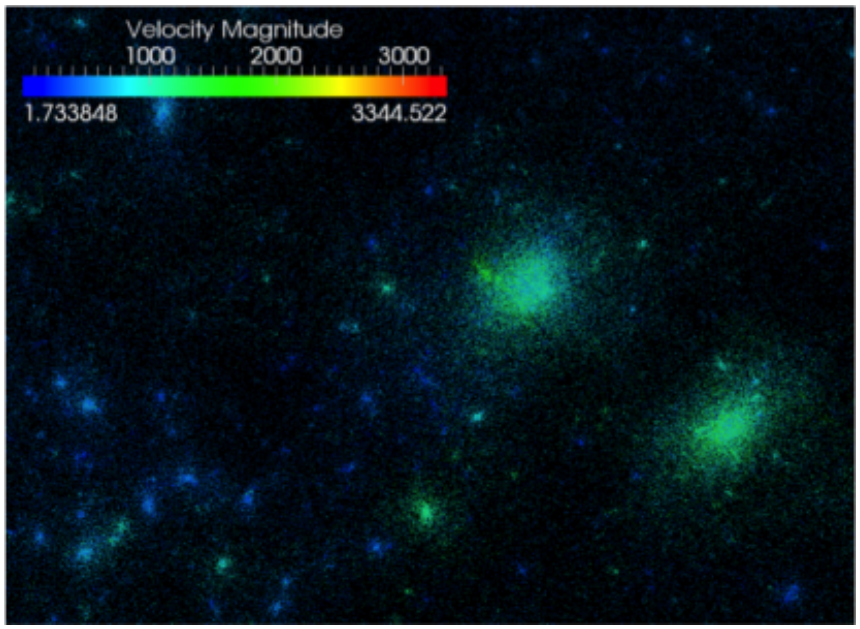
- Problem: irregular use pattern
- Solution: **burst-out** by **renting** resources on the Cloud for peak usage

- More expensive than local resources
- Difficult to justify non-capital expenditures

High Performance Computing

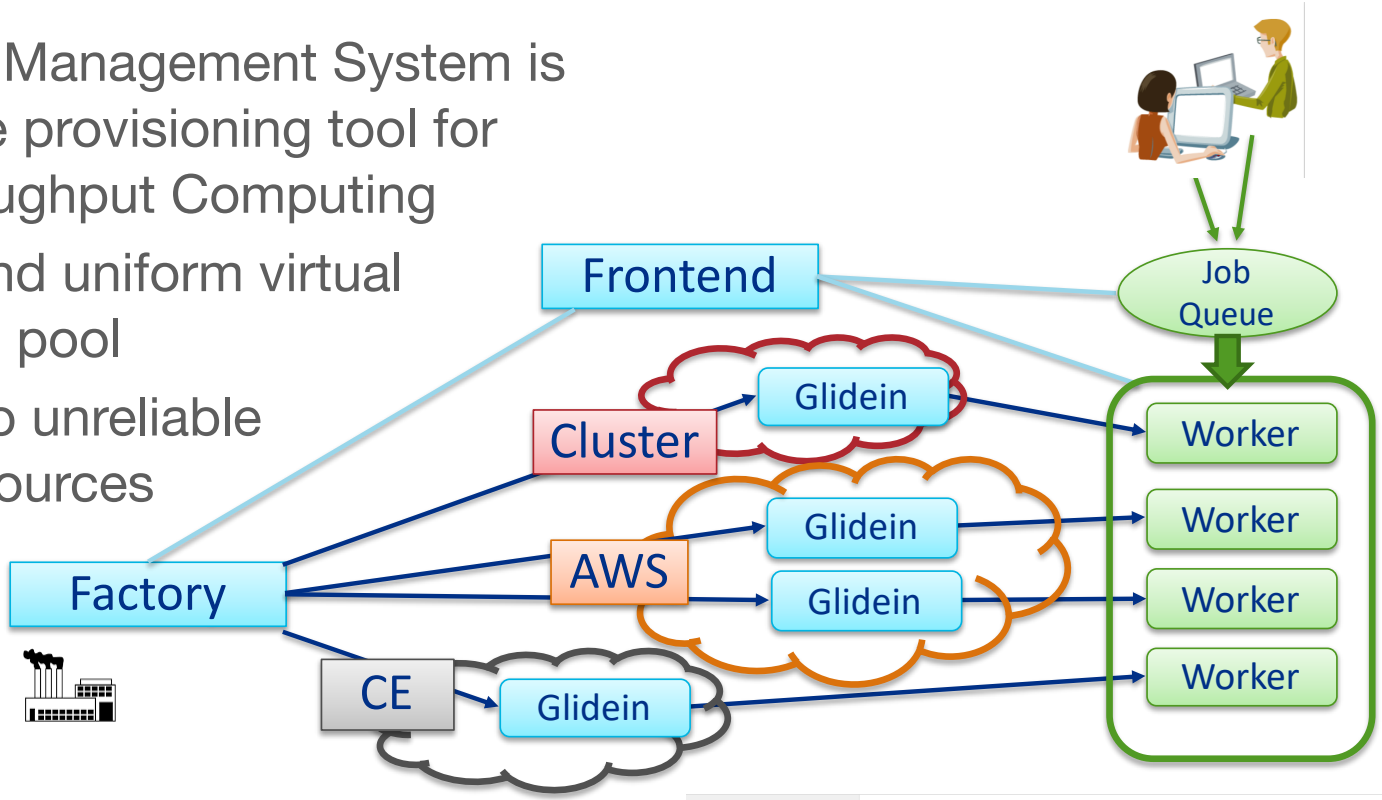
- Much scientific computing outside of experimental HEP is not “pleasantly parallel”
- Better platform: **High Performance Computing** (HPC)
  - Batch systems where individual computers are **interconnected** via high-speed links
  - Large HPC systems are often called “supercomputers”
- Fermilab has 5 HPC clusters
  - Total of 18.5k CPU cores
  - Used for Lattice QCD calculations, accelerator modeling, and large-scale astrophysical simulations
- Used also for event processing by splitting it into pieces

Mira@Argonne: 768k processors, 10 petaflops

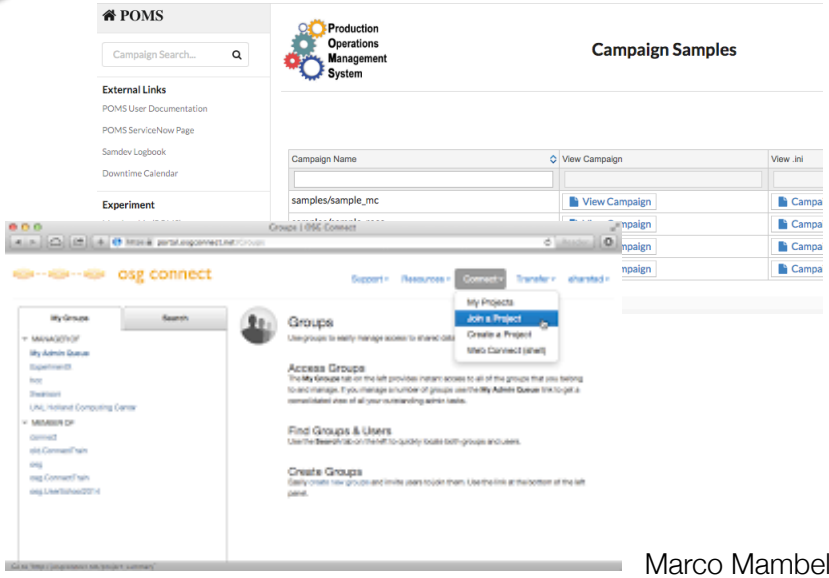


Putting it all together: GlideinWMS

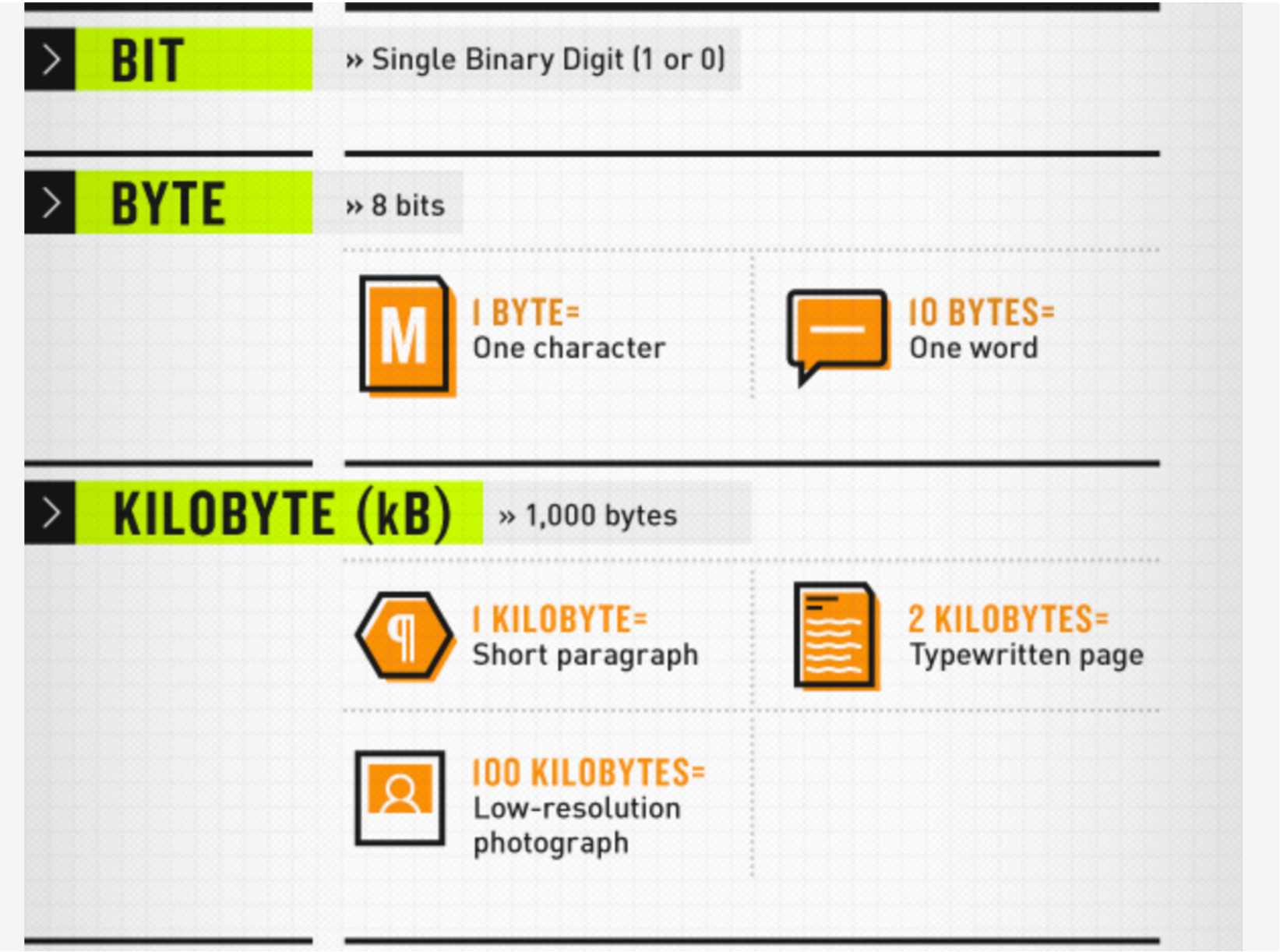
- The Glidein Workload Management System is a pilot based resource provisioning tool for Distributed High Throughput Computing
  - Provides reliable and uniform virtual clusters, the global pool
  - Submits Glideins to unreliable heterogeneous resources



- Knows "how to talk" to all the different systems
- Multiple Frontends and Factories work together to provide High Availability
- Used by: CMS Production and CRAB, POMS, Jobsub, OSG-Connect



Storage: understanding units



https://datascience.berkeley.edu/big-data-infographic/

MEGABYTE (MB) » 1,000 Kilobytes

1 MEGABYTE=
Short novel

2 MEGABYTES=
High-resolution photograph

5 MEGABYTES=
Complete works of Shakespeare

10 MEGABYTES=
Digital chest X-ray

100 MEGABYTES=
Two encyclopedia volumes

700 MEGABYTES=
CD-ROM

GIGABYTE (GB) » 1,000 Megabytes

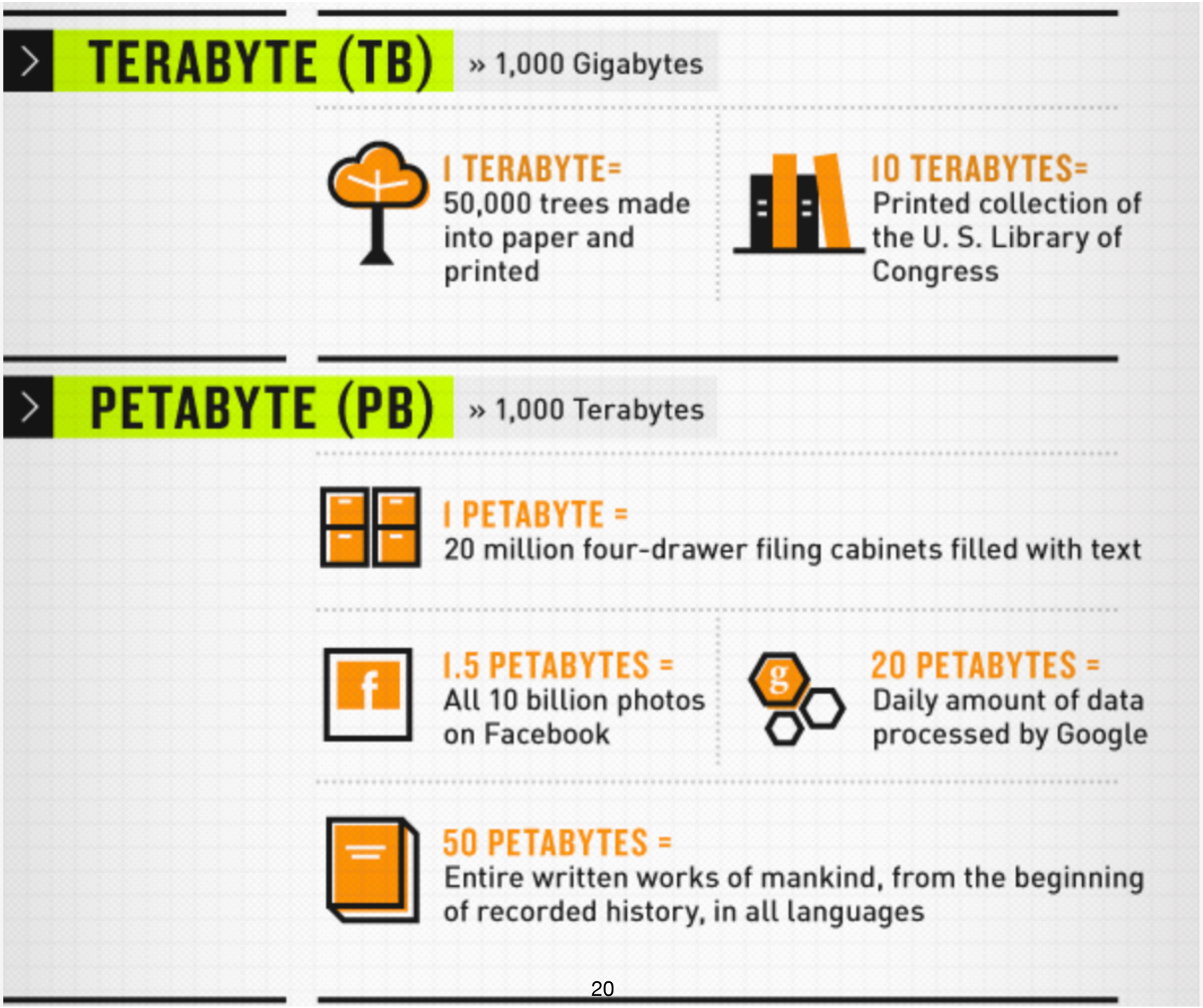
1 GIGABYTE=
7 minutes of HD-TV Video

4.7 GIGABYTES=
Size of a standard DVD-R

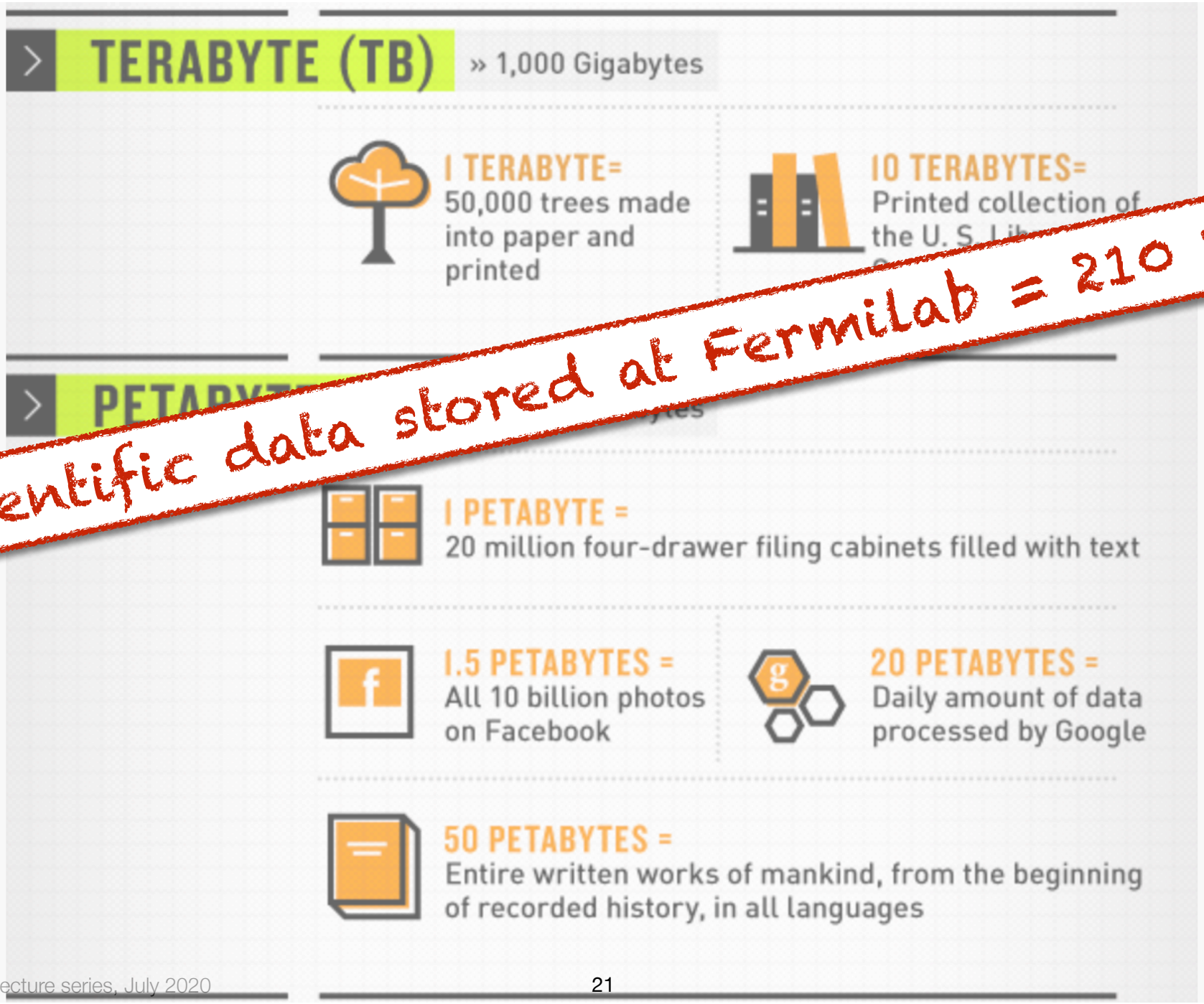
20 GIGABYTES=
Audio set of the works of Beethoven

100 GIGABYTES=
Library floor of academic journals

Storage: understanding units

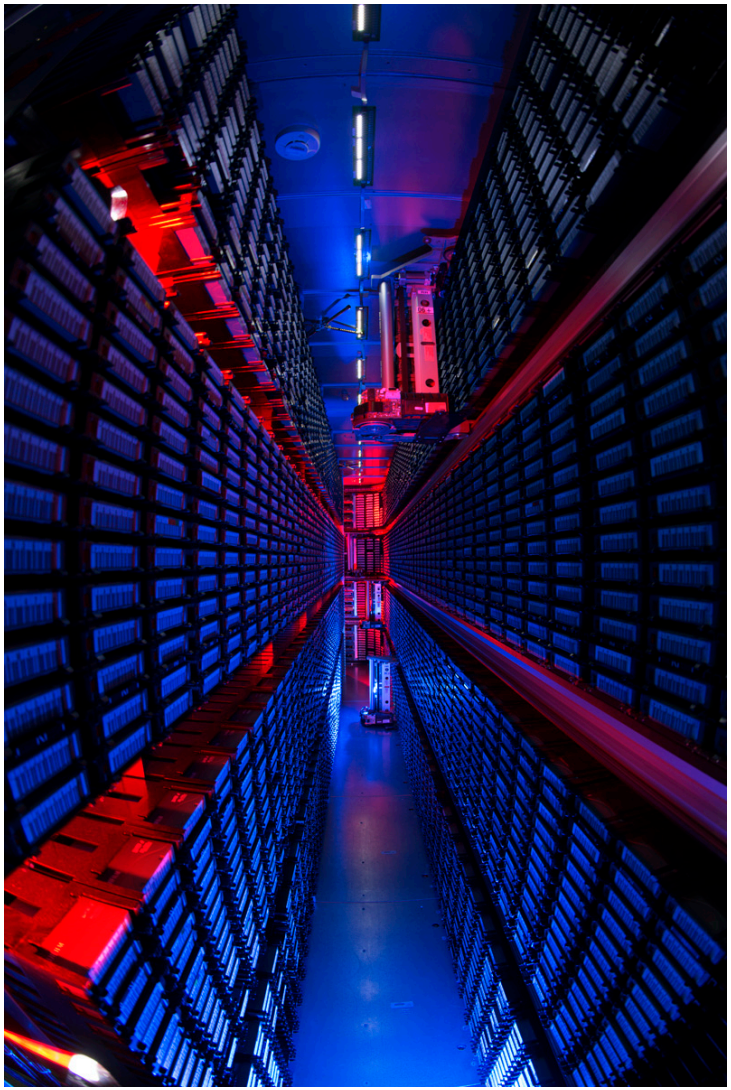


Storage: understanding units



Scientific data stored at Fermilab = 210 PB!

Storage: tape



- Primary storage medium for scientific data at Fermilab: **magnetic tape**
- Still the most efficient way to store petabytes of data if:
  - Not all of it is accessed at the same time
  - Access patterns are fairly linear
  - Sufficient disk for **staging**
- Fermilab has seven **robotic tape libraries**
  - Each library can hold 10,000 tapes
- Current tapes hold ~10TB of data each

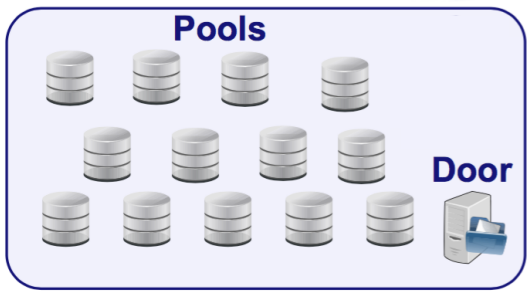
- Total active on tape: 164.07
  - cms, 71.83
  - NOVA, 27.24
  - uBoone, 23.53
  - gm2, 11.58
  - DUNE, 9.50
  - Plus other experiments and collaborations

Storage: disk

Harddrive

Diskpool

Storage System

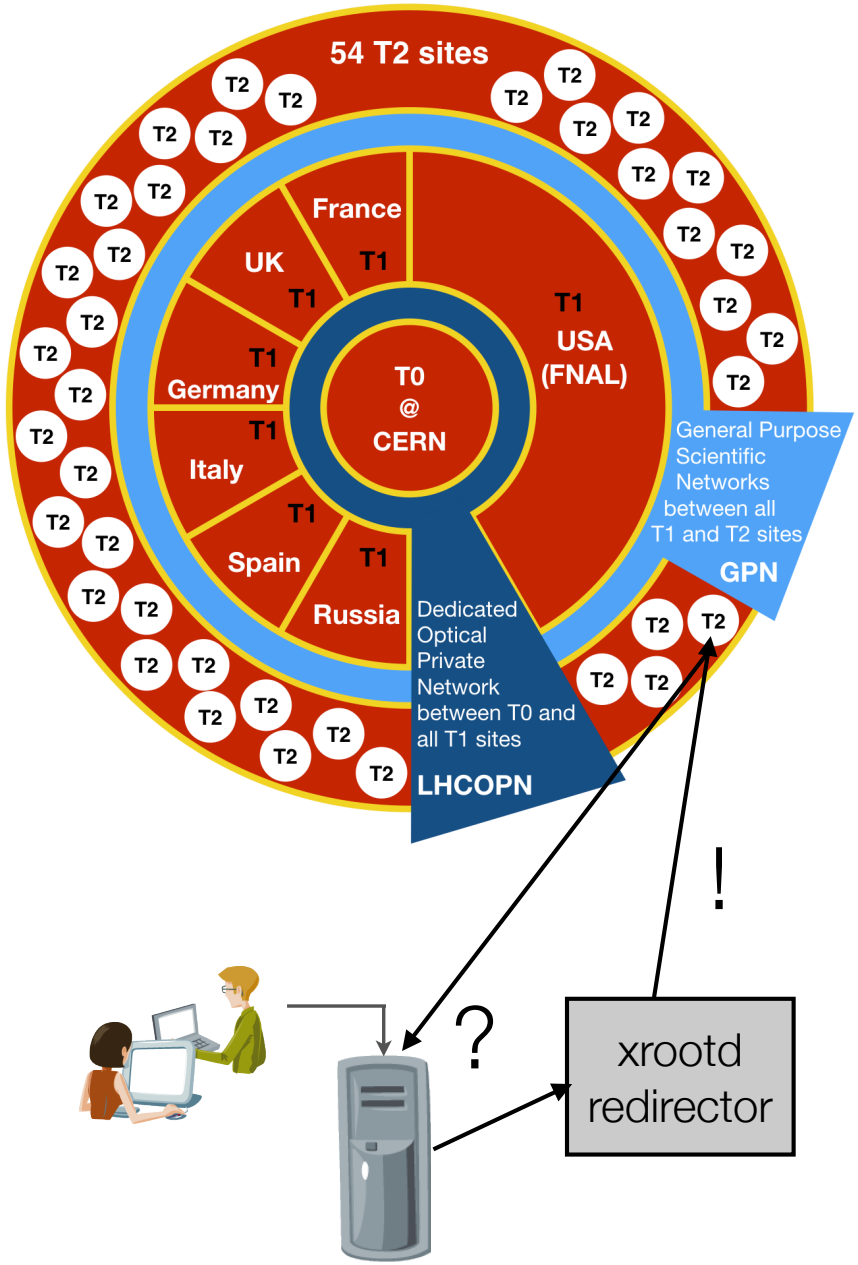


- ~46PB of disk (hard drive) storage
  - Most used as staging area from tape
- Disks are organized into **pools**
- Software allows collections of pools to appear to a user as a **single storage device**
  - Fermilab uses a system called “dCache”
  - In a typical week, data throughput in the Fermilab dCache pools average **30GB/s**

Federating data

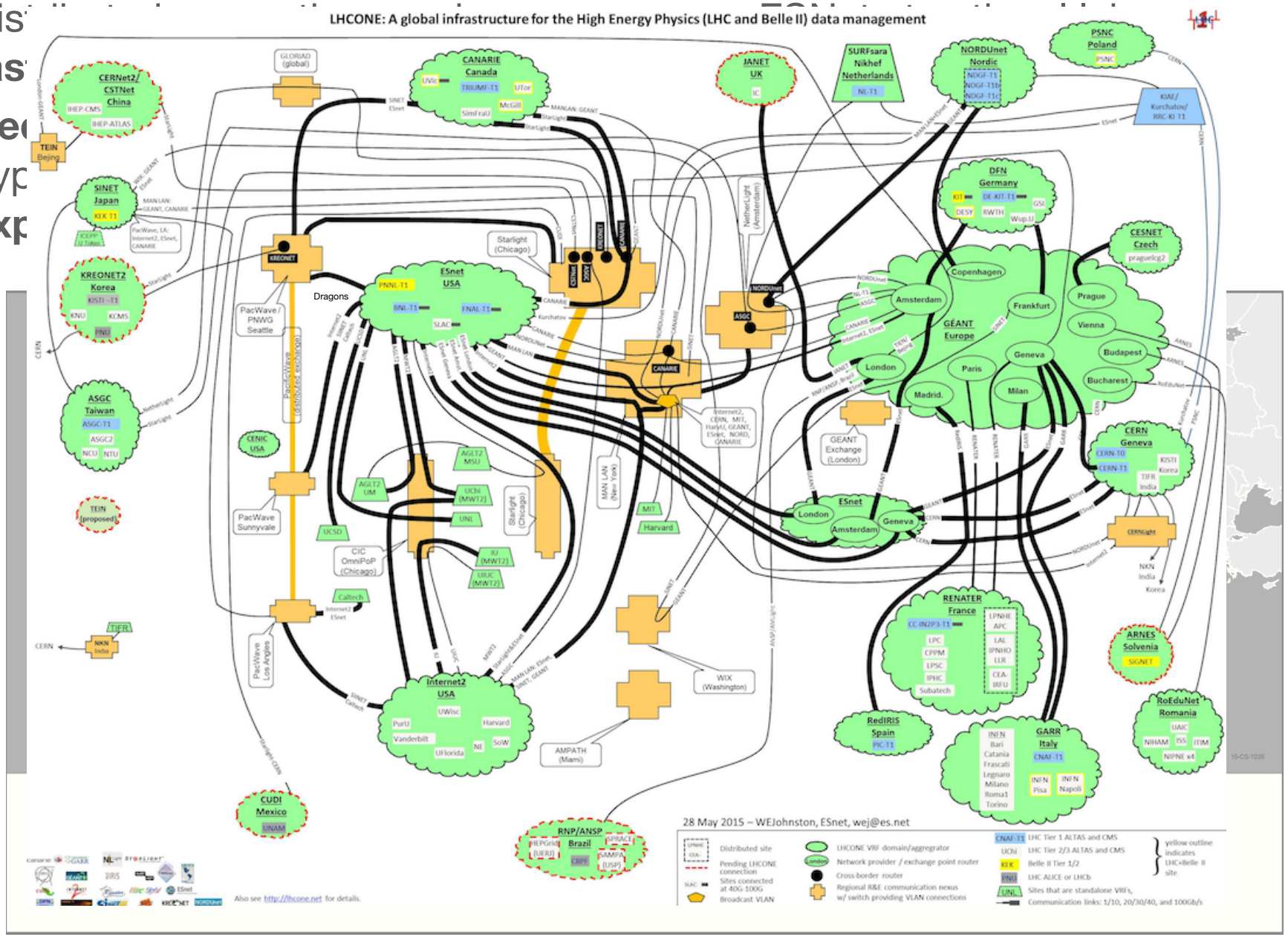
- CMS distributes data as well as CPU across grid sites
  - “Tier-0” is CERN
  - “Tier-1” sites in 7 countries
  - “Tier-2” sites at universities and labs
- Allows distribution of data copies across sites
- Data placement across sites can be automated

- **Redirection services** (xrootd) allow for automatic retrieval (streaming) of files
  - The user does not need to know the exact location of the file



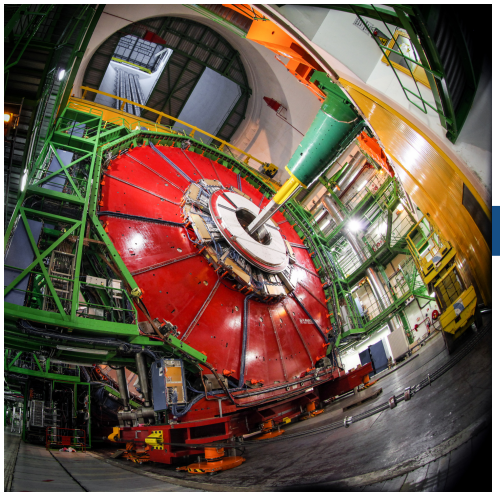
Networking

- Dis
  fas
- **De**
  (typ
  exp



Software

Reconstruction



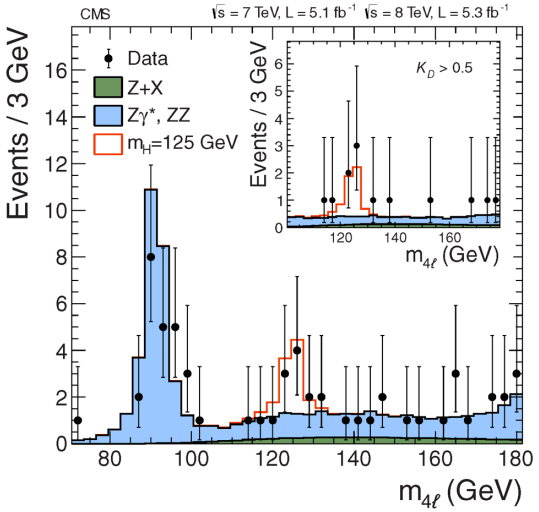
"Raw" data

Reconstructed data

Reduced data

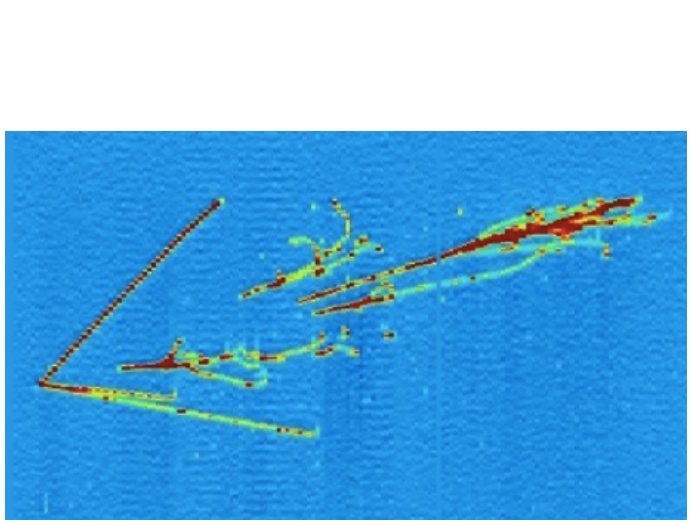
(further) Reduced data

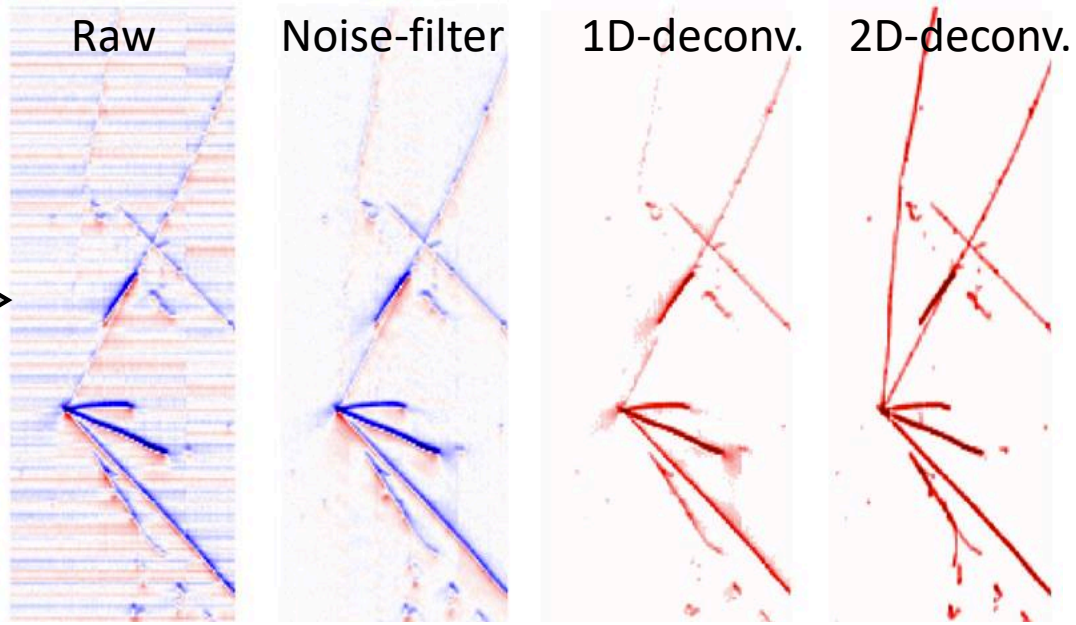
Monte Carlo simulation

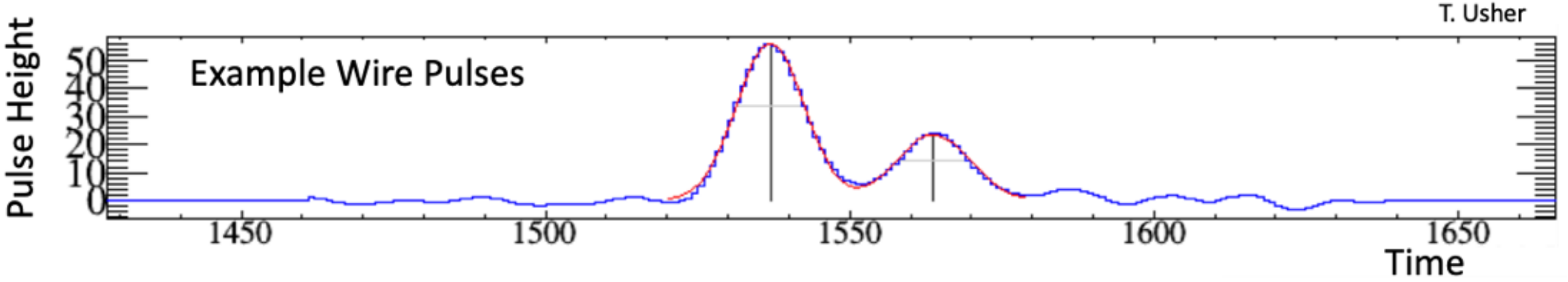


Reconstruction: LArSoft (MicroBooNE)

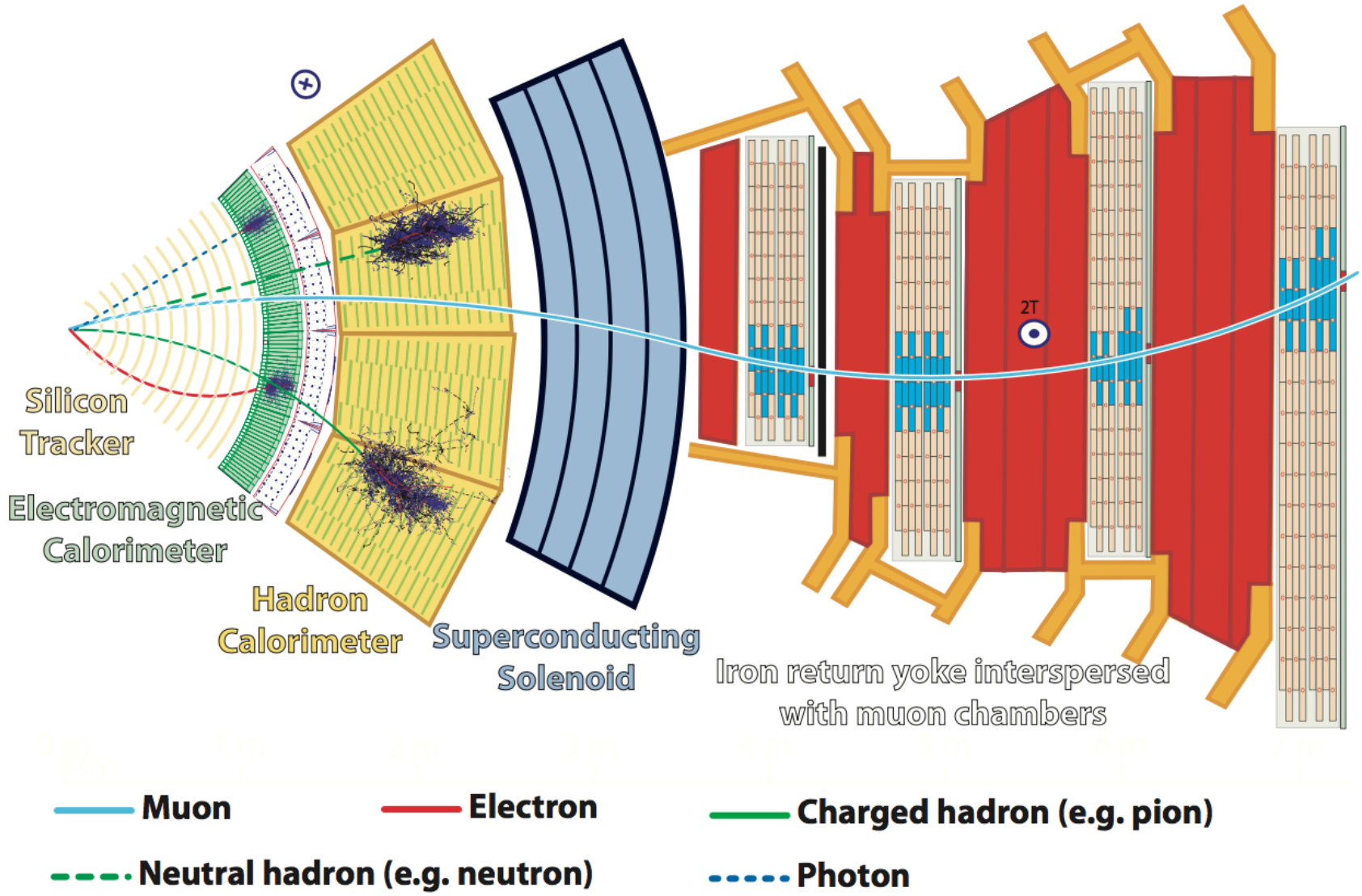
- **LArSoft**, developed at Fermilab, is a common code base for LAr neutrino detectors
  - Used by MicroBooNE, SBND, ICARUS, DUNE
  - Includes tools for both reconstruction and data analysis



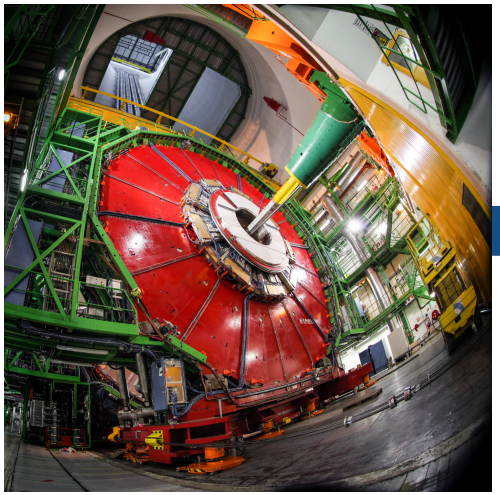




Reconstruction: CMS



Simulation

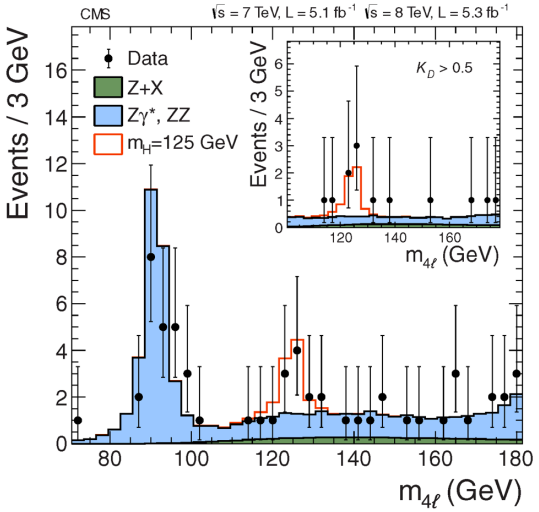


"Raw" data

Reconstructed data

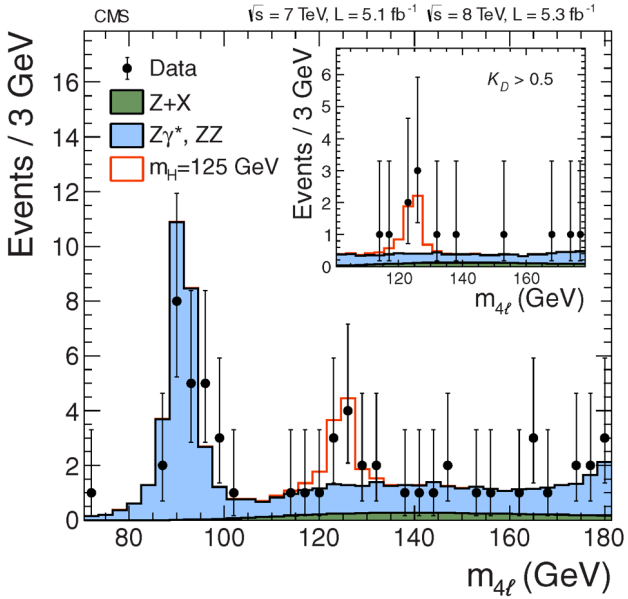
Reduced data

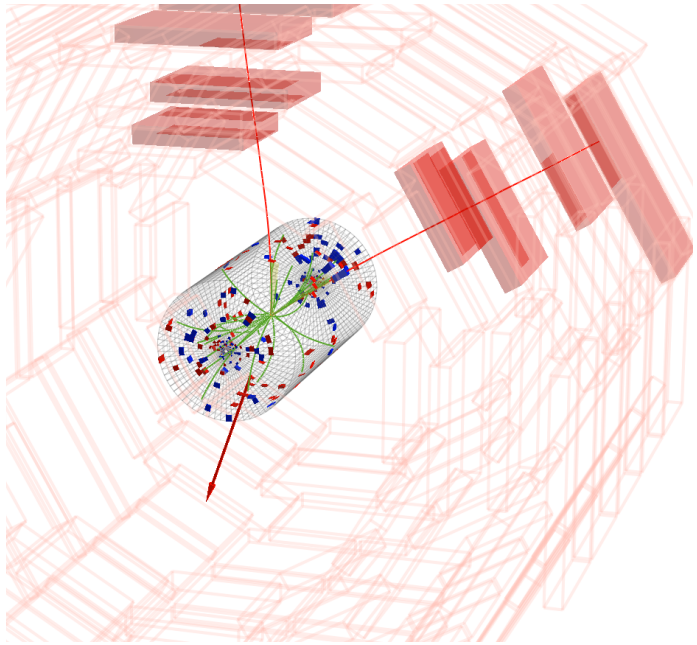
(further) Reduced data



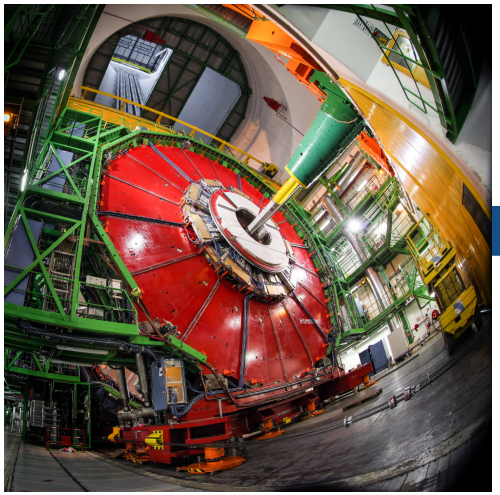
Simulation

- **Simulated events** are crucial to science at HEP experiments
  - Rely on **Monte Carlo Simulation** to produce these events
- **Event generators** simulate the underlying particle interaction of interest
- Resulting interaction event is then fed to a **detector simulator**
  - Consider the **material** and **geometry** of every part of the detector
  - Simulate how particles from interaction and decay would propagate
  - Most detector simulations use **GEANT**
    - Also used in nuclear and accelerator physics as well as medical and space science





Analysis



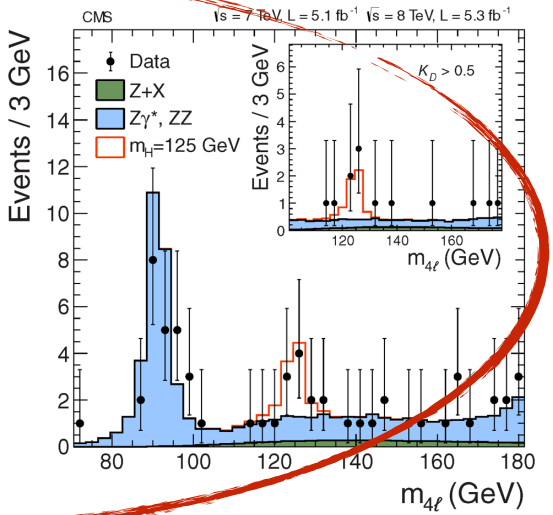
"Raw" data

Monte Carlo simulation

Reconstructed data

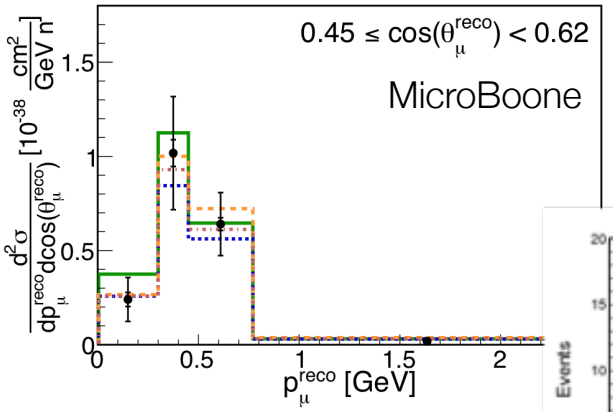
Reduced data

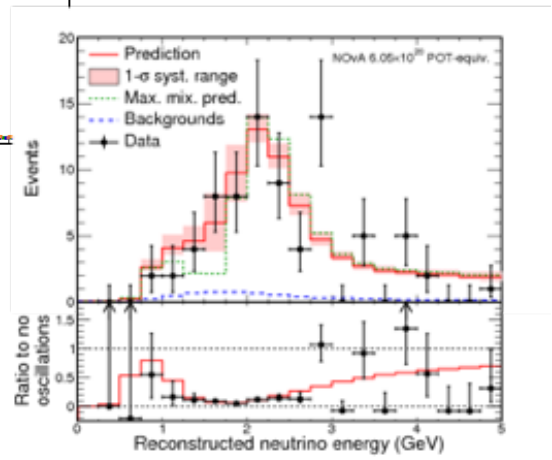
(further) Reduced data

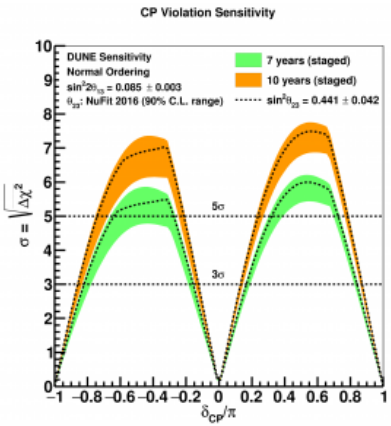


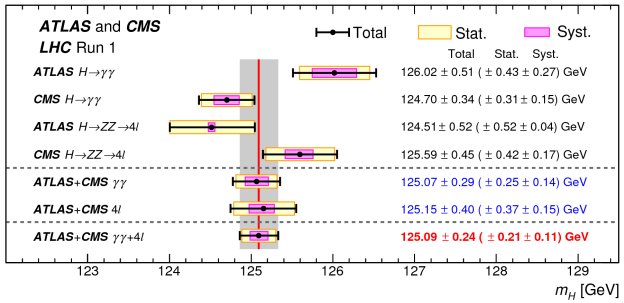
Analysis

- Getting from data events of interest to plots, tables, and numbers
  - This is the computing step nearly all HEP experimentalists are familiar with
- Common tools are needed
  - Mathematical functions
  - Statistical analysis
  - Plotting/histogramming
- Nearly all HEP experiments use the **ROOT** framework
  - Developed by CERN and Fermilab
  - **C++** (object oriented)
  - Couples with code written in other languages (e.g. Python)





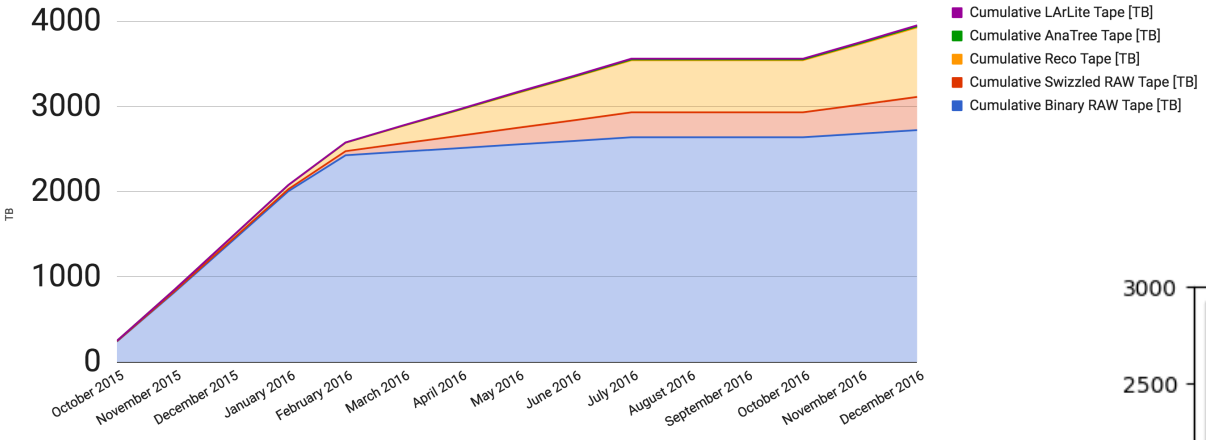




The Future

An explosion of data

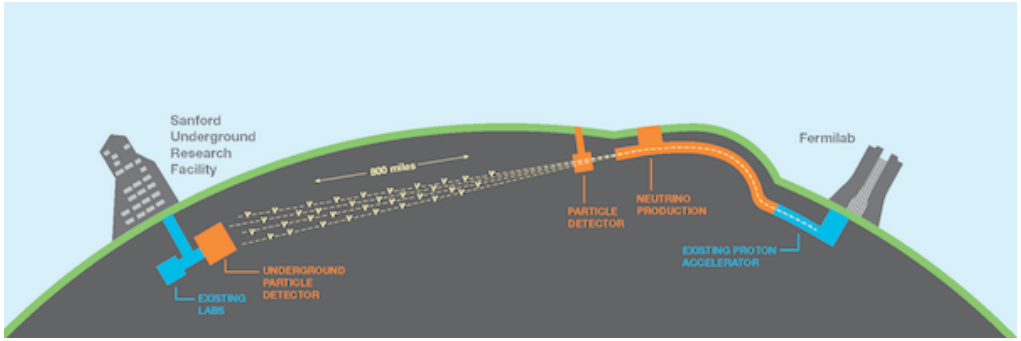
Data Volume [TB]

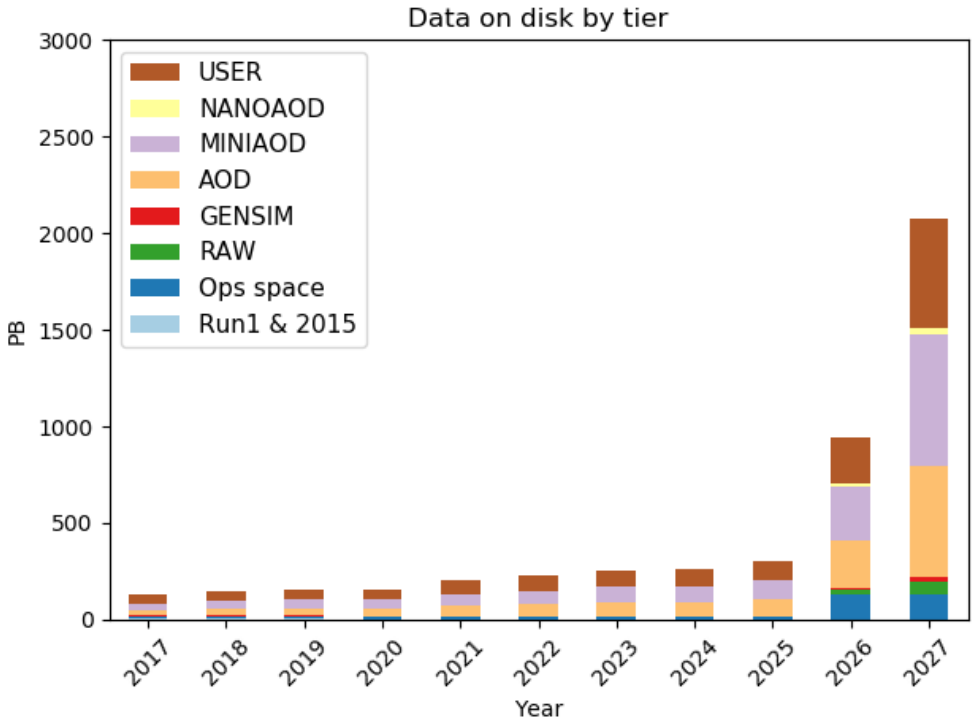


Month

MicroBooNE tape usage in ~1 year

DUNE (~2026) will produce **> 50 PB** of data a year





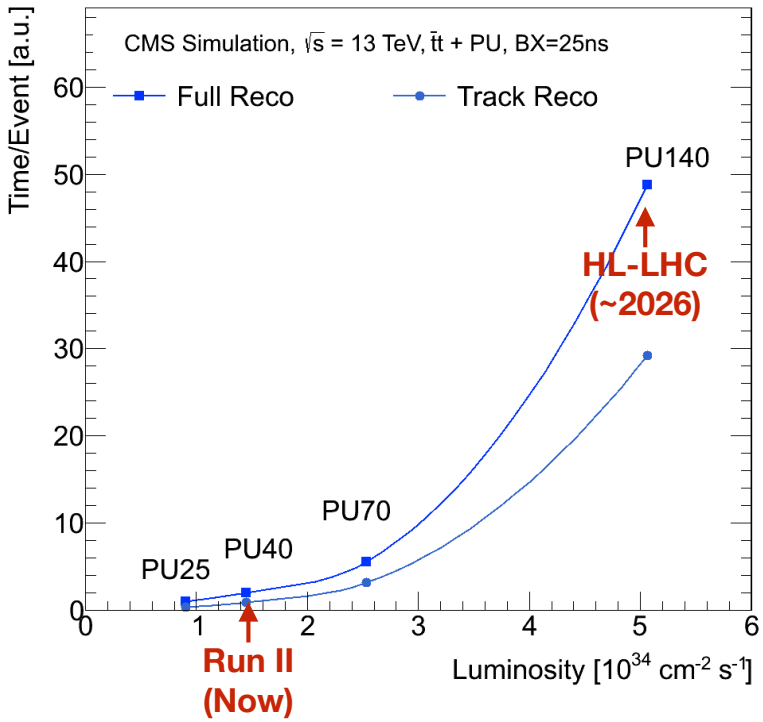
Expected LHC storage needs for Run 4 = **Exabytes**!

Ever-growing need for CPU

- Growing **dataset size** and **event complexity** = **more computing!**
  - If we scale **current algorithms**, the CPU needs of LHC experiments will grow by a factor of **30** in a decade
  - Similar issues faced by newer, bigger LAr experiments such as DUNE

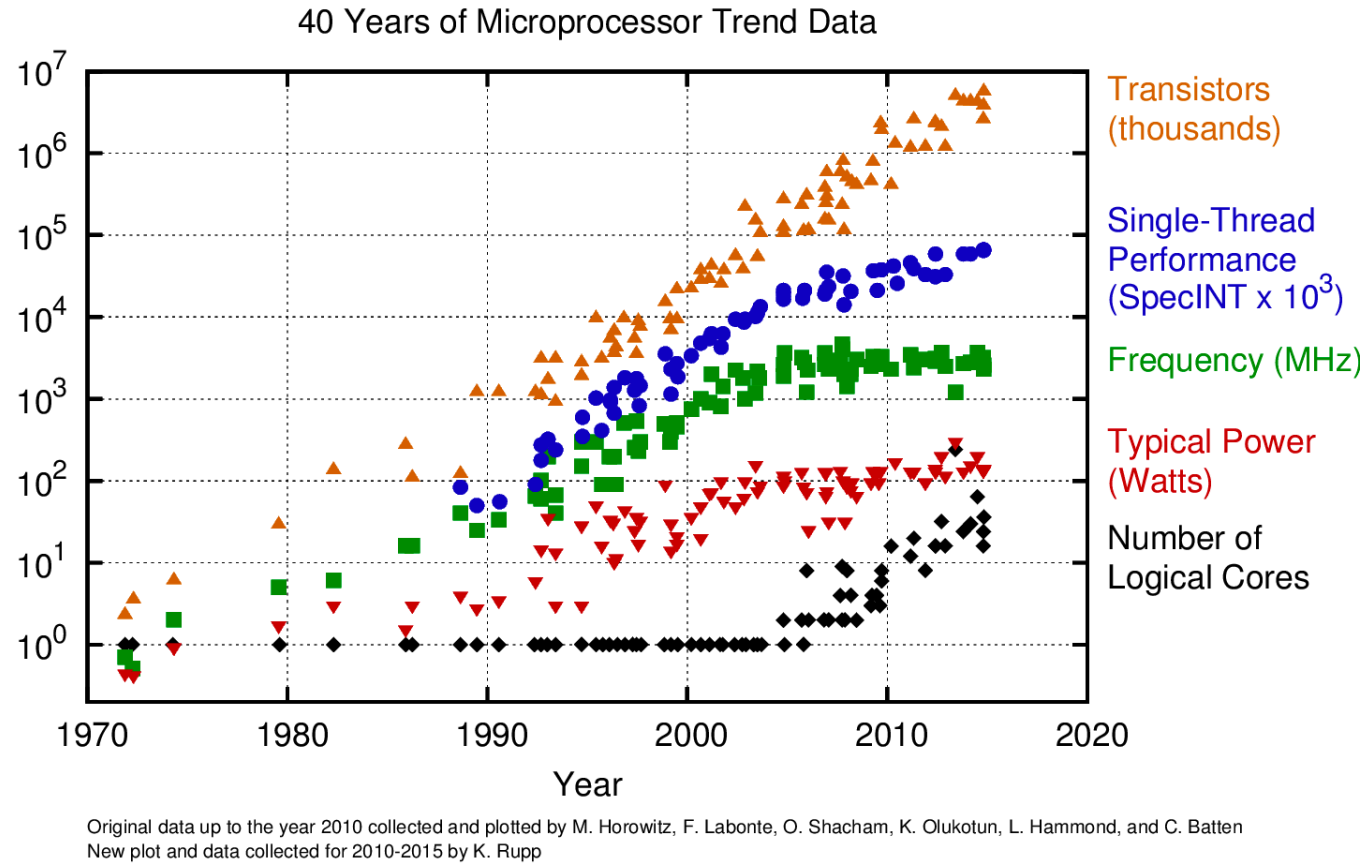
- In the past **technology improvements** have helped HEP keep up with computing demands
  - Expected technology improvements alone will only get us **1/6th** of the way there

- What about **Moore's Law…**?



Moore's Law probably won't help

- Moore's Law: **Number of transistors** on a chip doubles every 2 years
- That's still true, but **single-thread performance** has stopped increasing
- Instead, **number of cores** is now dramatically increasing



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

To take advantage of these:

Intel Many Integrate Core (MIC) CPU

NVIDIA GPU

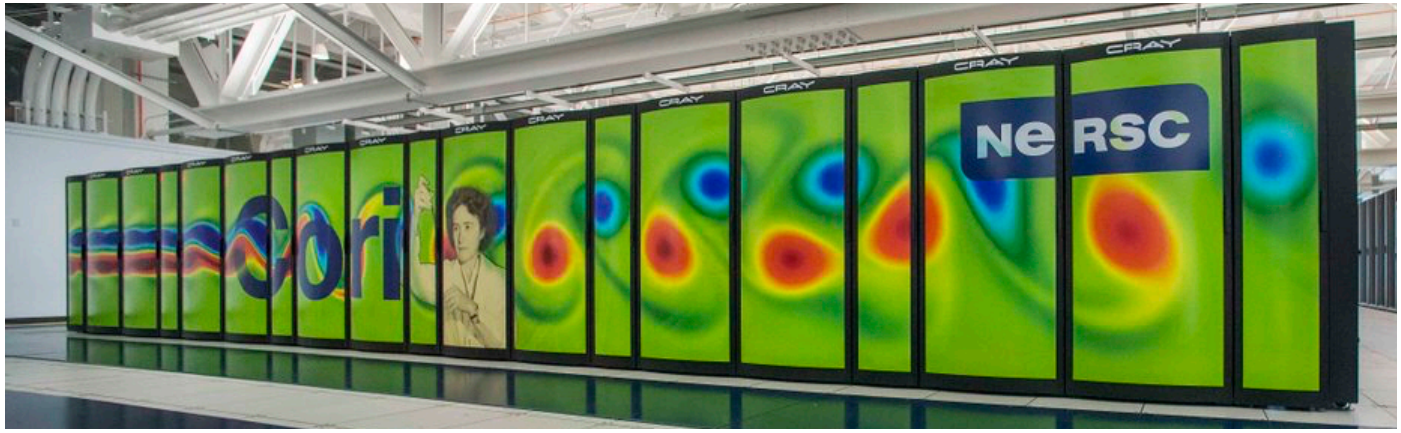
We need to rewrite a lot of software!

What about supercomputers?

ASCR* Computing Upgrades At a Glance

System attributes	NERSC Now	OLCF Now	ALCF Now	NERSC Upgrade	OLCF Upgrade	ALCF Upgrade
Name	Cori	Summit	Theta	Perlmutter	Frontier	Aurora
Installation (planned or actual)	2016	2018	2017	2020	2021	2021
System peak (PF)	30	200	12	~100	1500	> 1000
Peak Power (MW)	3.9	13	1.7	?	?	?

This is the hard part - getting the computing power without melting the building.



*Advanced Scientific Computing Research (Dept. of Energy)

PF = petaflops, floating point operations per second

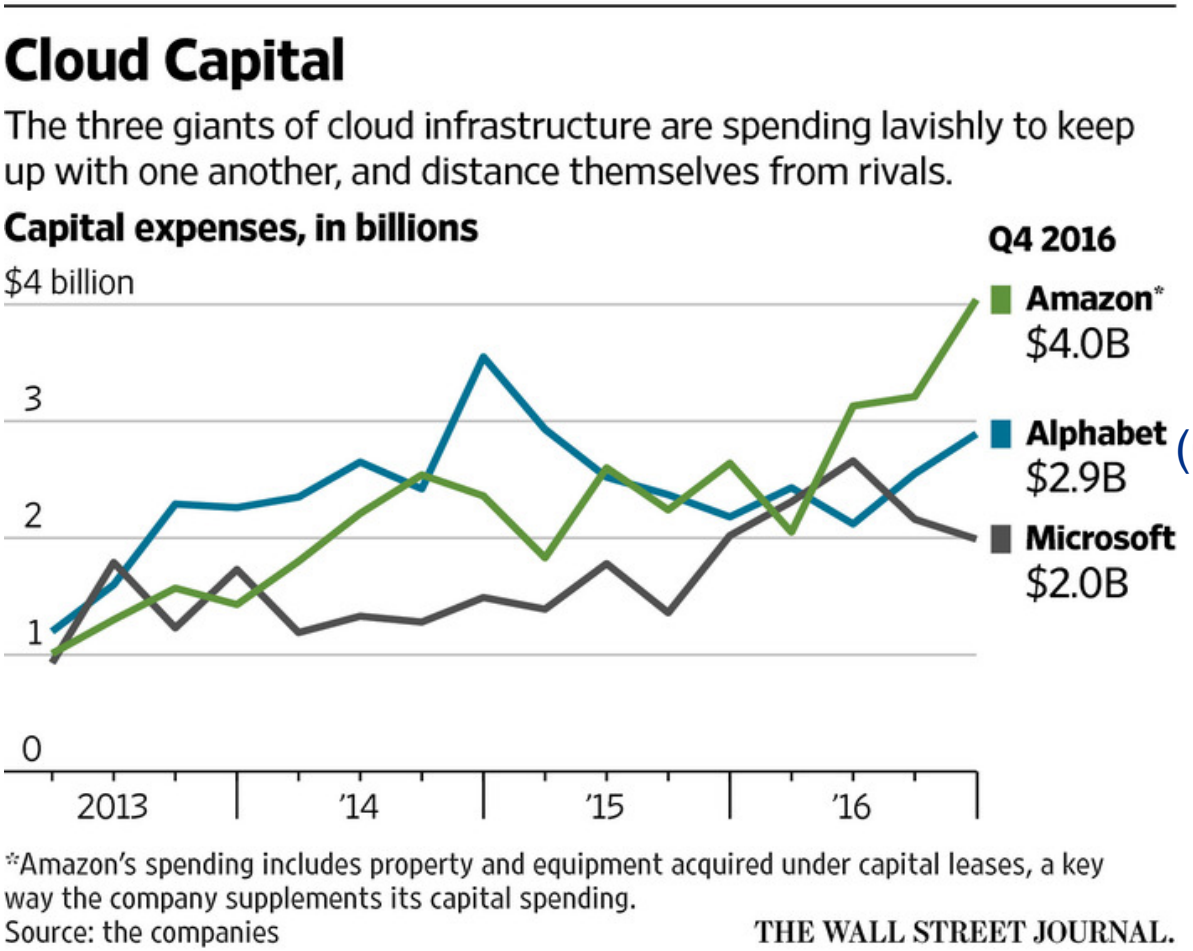
1,000 PF = 1 exaflops

"Excascale" (10^{18}) by 2021

http://exascaleproject.org/

Commercial cloud computing (>> HEP computing)

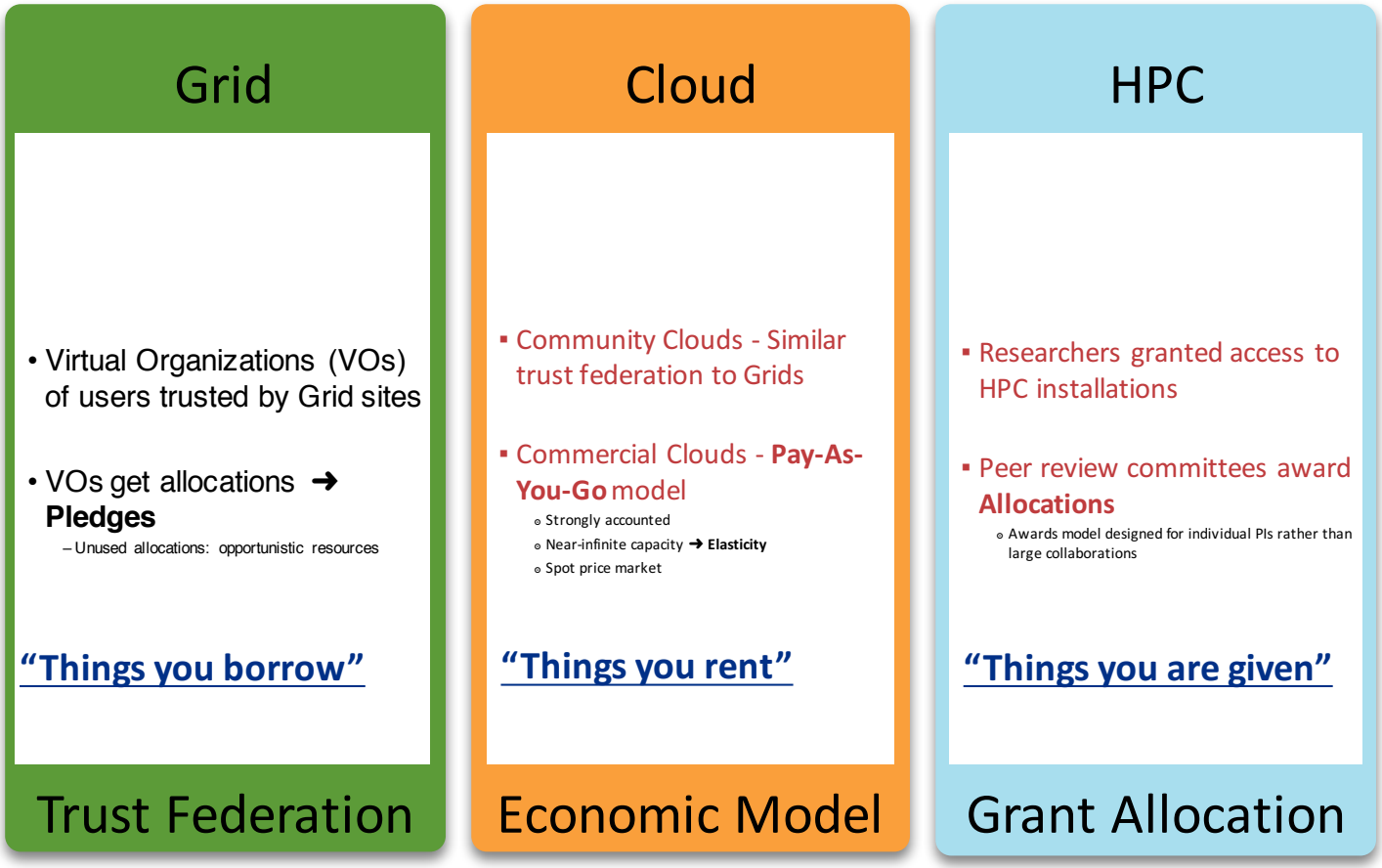
- Total spending on cloud computing is now > $200 billion per year
- Many huge companies (Netflix, for example) don't buy their own clusters but rely entirely on cloud computing
- HEP experiments are using these resources as well



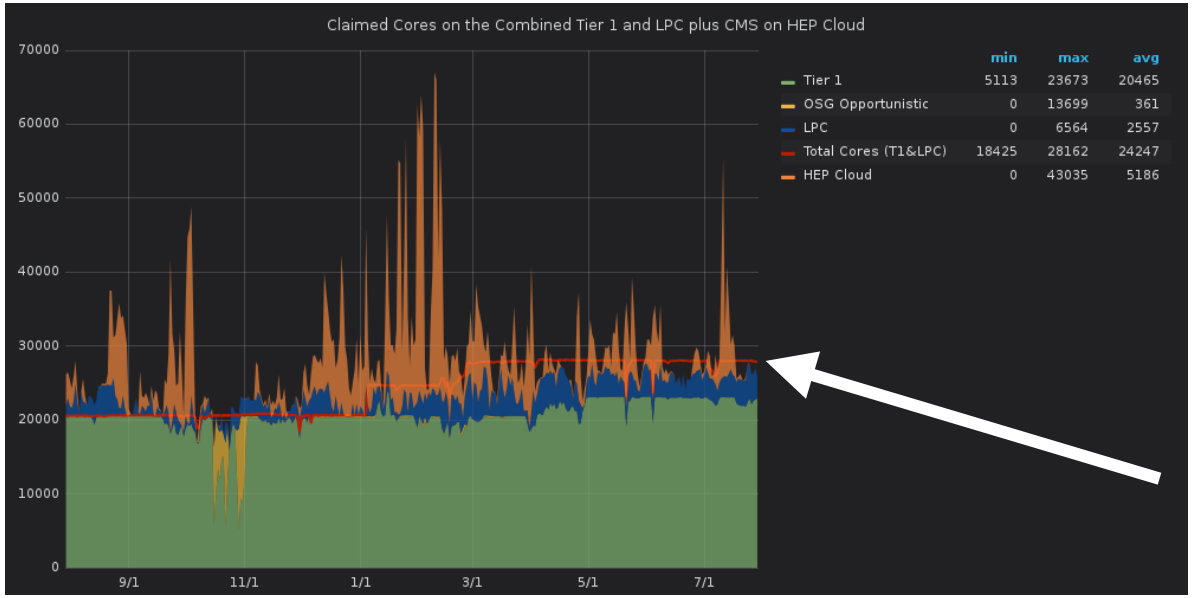
(Google)

HEPCloud

- Making it all possible from one place: **HEPCloud**
- **Unified interface** to Grid, Cloud, and HPC resources
- Currently being used to run CMS workflows on NERSC supercomputers



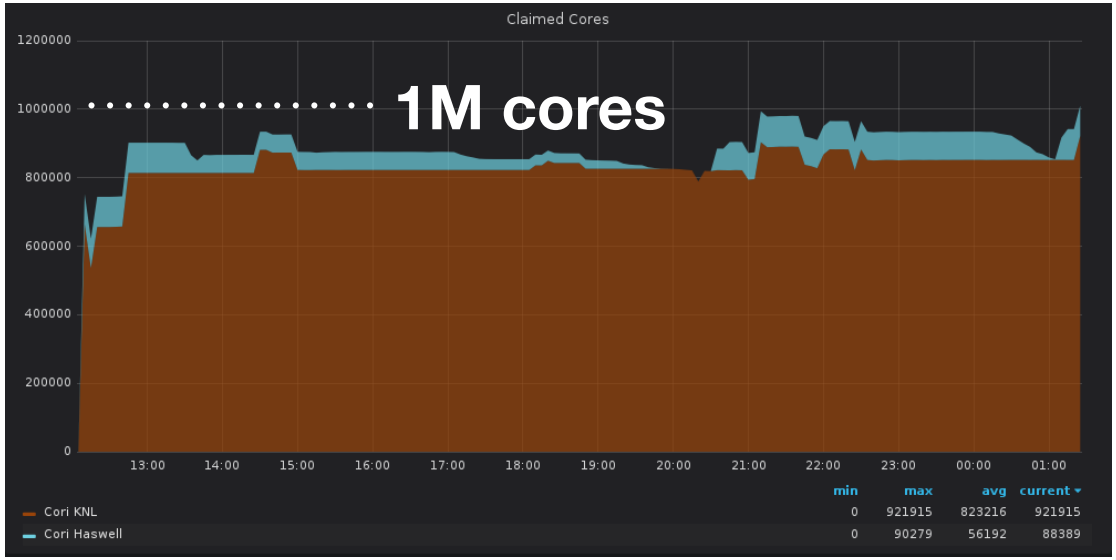
HEPCloud: showing it works



Doubling CMS computing capacity using HEPCloud to run on NERSC!

Simulate 1 billion events in 48 hours

Orange peaks represent HEPCloud usage



NOvA running at NERSC

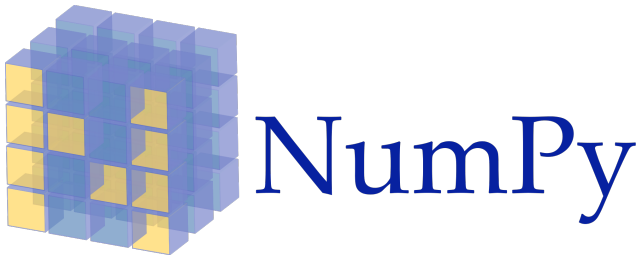
Over 1 million cores used simultaneously

Analysis techniques: tools from industry

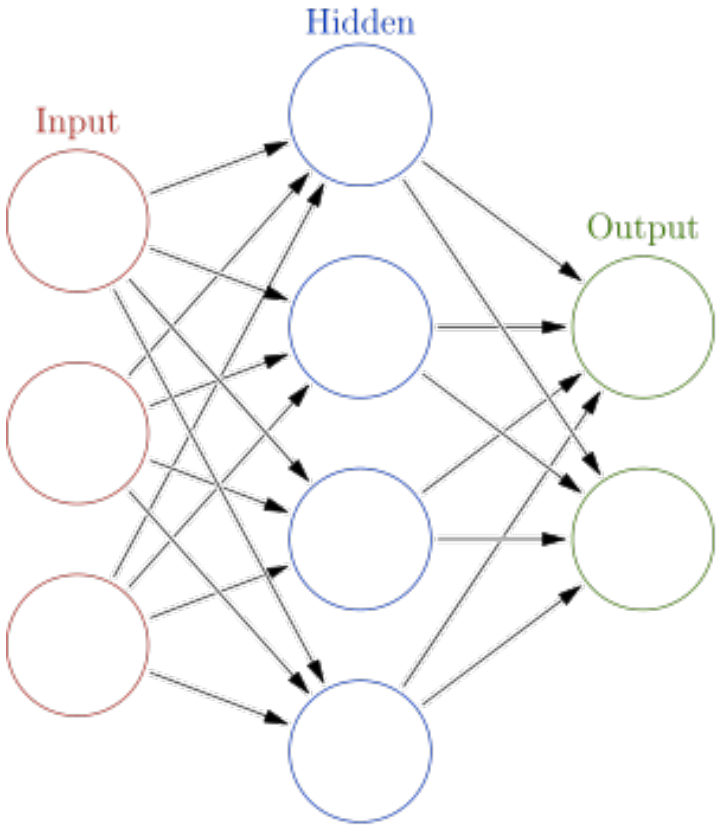
- HEP experiments were some of **the first** cases where people had to deal with analyzing really big datasets
  - Had to develop our own tools to get the science done (ROOT, for example)
- Not true anymore. Basically every big company you can think of has huge amounts of data at their fingertips
  - Many tools of been developed outside labs and universities to help store, process, and analyze all this data

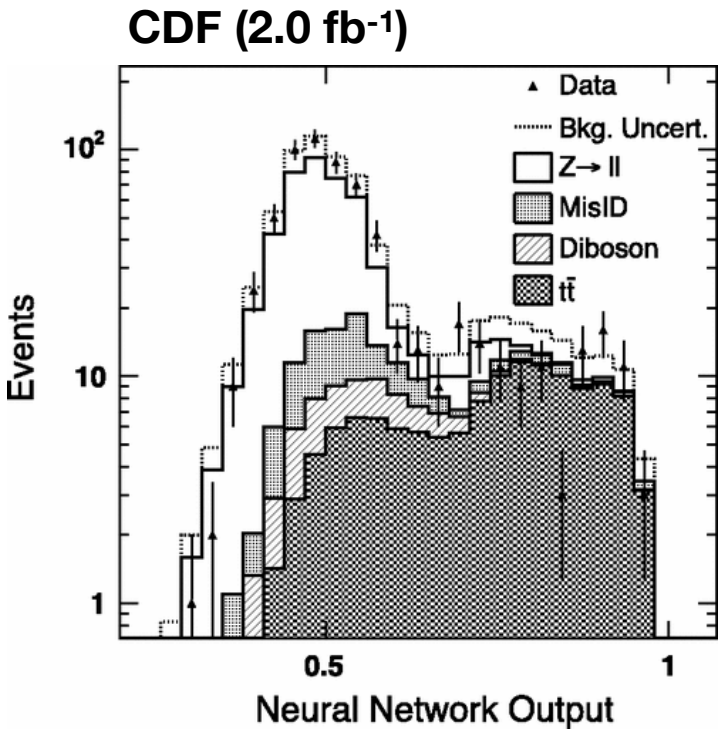
- Fermilab's approach for CMS analysis is COFFEA (the COmpact Framework for Elaborate Algorithms)

  - Instead of a **for loop** over events, use **array programming** expressions to process many events simultaneously
  - Uses Apache Spark and tools from the scientific python "ecosystem" based on numpy



Analysis techniques: machine learning

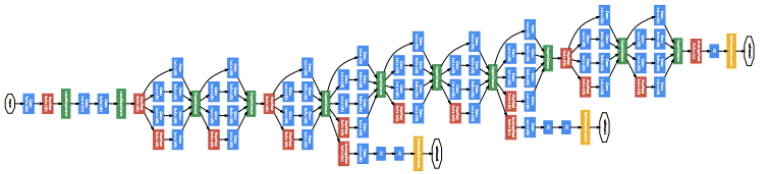


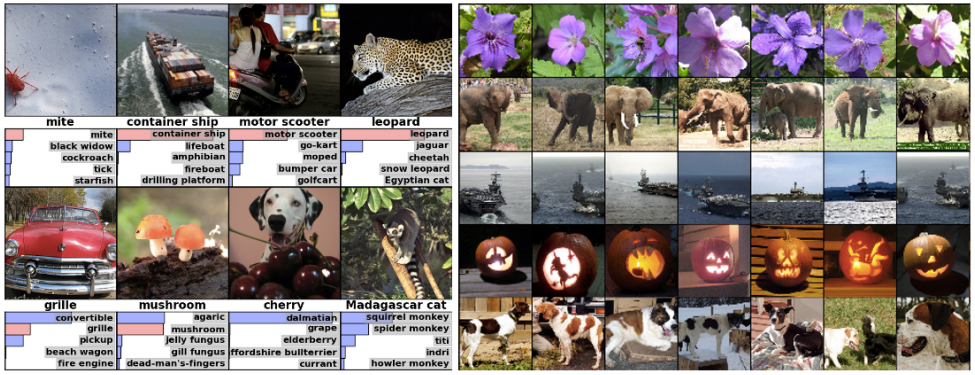


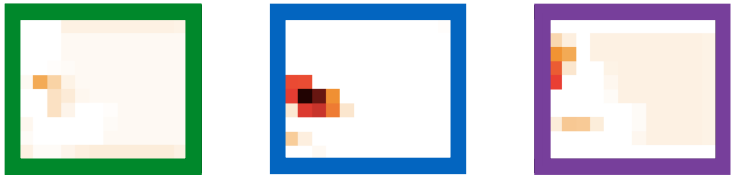
- Machine learning techniques (e.g., NNs) in use in HEP and Astrophysics since the turn of the century.

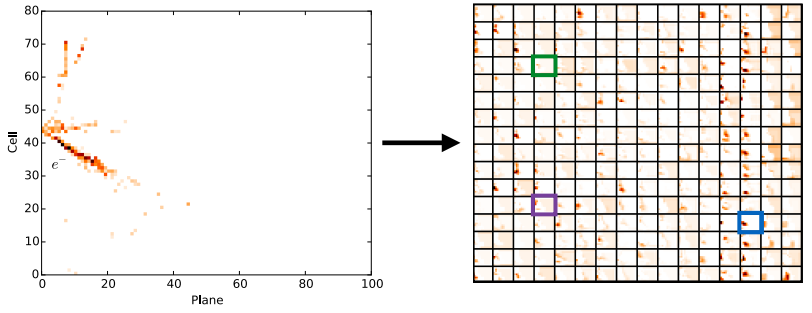
Analysis techniques: deep learning

- Deep learning **adds layers** (complexity) to the network
  - Can more effectively find features (especially in 2D+ inputs)
  - Widely used in computer **vision**, speech recognition, and other applications
- Particle physics events can essentially be considered as images
- DNNs are now being applied to event analysis, such as in **NOvA**
  - Allows finding features in hit maps (events) that aren't easily seen conventionally
  - **~30% improvement** even event ID
- See talk by Brian Nord on 6/25 for more









Conclusions

- The complexity of HEP experiments doesn't stop with the detectors
- Scientific computing permeates every aspect of how we do physics at Fermilab

- There are challenges ahead
  - Many that **you** could help solve!

- **Thanks to Allison Hall for letting me use her slides**

- **Thanks to all those who helped with content**
  - Especially: Dmitry Litvintsev, Sophie Berkman, Oliver Gutsche, Ken Herner, Burt Holzman, Michael Kirby, Anne Schukraft, Erica Snider, Alexander Radovic

Questions?

http://computing.fnal.gov